

DIALOG(R)File 351:Derwent WPI  
(c) 2006 The Thomson Corporation. All rts. reserv.

0002822834

WPI ACC NO: 1983-C5117K/

**Re-structural integrated circuit with high gate density - has  
multiprocessor operated as lock-step or pipelined unit, and multilevel  
interrupt management system**

Patent Assignee: TEXAS INSTR INC (TEXI)

Inventor: BUDZINSKI R L

**Patent Family (4 patents, 5 countries)**

Patent Number	Kind	Date	Application Number	Kind	Date	Update
EP 71727	A	19830216	EP 1982105491	A	19820623	198308 B
JP 58058672	A	19830407			198320	E
EP 71727	B	19861001	EP 1982105491	A	19820623	198640 E
DE 3273549	G	19861106			198646	E

Priority Applications (no., kind, date): US 1981286426 A 19810724; US  
1981286425 A 19810724; US 1981286424 A 19810724

#### Patent Details

Number	Kind	Lan	Pg	Dwg	Filing	Notes
EP 71727	A	EN	28			

Regional Designated States,Original: DE FR GB NL  
EP 71727 B EN  
Regional Designated States,Original: DE FR GB NL

#### Alerting Abstract EP A

The circuit has a monolithic substrate, having a number of 16-bit processors all accessing a common memory, busses including a status bus with synchronisation and arithmetic linkage lines. Each processor comprises a respective status bus multiplexer connected to the respective processor status bus. A bus control unit selectively connects the processor to the data bus or directly to a corresponding RAM memory module.

Each processor also contains interrupt manager which tests each successive interrupt signals to determine whether the respective processor which includes the interrupt manager is designated by the successive interrupt signal as an interrupter. The interrupt manager stores the priority of the one of the interrupt signals which initiated respective sequence of commands is currently being executed by the processor. The processors may be reconfigured programmably to operate independently in lock step or as pipelined processors.

**Title Terms /Index Terms/Additional Words:** STRUCTURE; INTEGRATE; CIRCUIT;  
HIGH; GATE; DENSITY; MULTIPROCESSOR; OPERATE; LOCK; STEP; PIPE; UNIT;  
MULTILEVEL; INTERRUPT; MANAGEMENT; SYSTEM

#### Class Codes

(Additional/Secondary): G06F-013/00, G06F-015/06, H01L-027/04

File Segment: EPI;

DWPI Class: T01

Manual Codes (EPI/S-X): T01-F01; T01-F02; T01-J02

#### Original Publication Data by Authority

**Germany**

Publication No. DE 3273549 G (Update 198646 E)

Publication Date: 19861106

Language: DE

Priority: US 1981286424 A 19810724

US 1981286425 A 19810724

US 1981286426 A 19810724

**EPO**

Publication No. EP 71727 A (Update 198308 B)

Publication Date: 19830216

**\*\*Integrierte Schaltung mit aenderbarer Struktur**

Restructurable integrated circuit

Circuit integre restructurable\*\*

Assignee: TEXAS INSTRUMENTS INCORPORATED, 13500 North Central Expressway,  
Dallas Texas 75265, US (TEXI)

Inventor: Budzinski, Robert L., 1106 Edgewood Drive, Richardson Texas  
75081, US

Agent: Leiser, Gottfried, Dipl.-Ing., et al, Patentanwaelte Prinz, Bunke  
Partner Ernsberger Strasse 19, D-8000 Muenchen 60, DE

Language: EN (28 pages)

Application: EP 1982105491 A 19820623 (Local application)

Priority: US 1981286424 A 19810724

US 1981286425 A 19810724

US 1981286426 A 19810724

Designated States: (Regional Original) DE FR GB NL

Original IPC: G06F-13/00 G06F-15/06 H01L-27/04

Current IPC: G06F-13/00 G06F-15/06 H01L-27/04

Original Abstract: Restructurable integrated circuit. A restructurable integrated circuit, including four 16-bit processors PR0, PR1, PR2, PR3, data and control memories 66 and 78, and external interfaces 72, 73, 74, 75, 76 all mounted on a chip. The processors include reconfigurable connections through a status bus 52, microprogramming capability with dynamic logic array interpretation, and a multi-level flexible interrupt management system, so that the processors PR0-PR3 may be reconfigured programmably to operate independently, in lockstep, or as pipelined processors. All processors PR0-PR3 are connected to data, control, and status busses 56, 14, and 52, in addition, external control, data, and status interfaces 72-76 are also provided, connected through the respective corresponding busses 56, 14, and 52 to each of the processors PR0-PR3. These external interfaces are connected to all of the interconnections which permit reconfigurability among the processors on a chip, and these external interfaces permit coordination of the processors on more than one RIC chip.

Publication No. EP 71727 B (Update 198640 E)

Publication Date: 19861001

**\*\*Integrierte Schaltung mit aenderbarer Struktur**

Restructurable integrated circuit

Circuit integre restructurable\*\*

Assignee: TEXAS INSTRUMENTS INCORPORATED, 13500 North Central Expressway,  
Dallas Texas 75265, US

Inventor: Budzinski, Robert L., 1106 Edgewood Drive, Richardson Texas  
75081, US

Thatte, Satish M., 1304 Elk Grove, Richardson Texas 75081, US  
Agent: Leiser, Gottfried, Dipl.-Ing., et al, Patentanwaelte Prinz, Bunke  
Partner Ernsberger Strasse 19, D-8000 Muenchen 60, DE

Language: EN

Application: EP 1982105491 A 19820623 (Local application)

Priority: US 1981286424 A 19810724

US 1981286425 A 19810724

US 1981286426 A 19810724

Designated States: (Regional Original) DE FR GB NL

Claim: The circuit has a monolithic substrate, having a number of 16-bit processors all accessing a common memory, busses including a status bus with synchronisation and arithmetic linkage lines. Each processor comprises a respective status bus multiplexer connected to the respective processor status bus. A bus control unit selectively connects the processor to the data bus or directly to a corresponding RAM memory module.

Each processor also contains interrupt manager which tests each successive interrupt signals to determine whether the respective processor which includes the interrupt manager is designated by the successive interrupt signal as an interrupter. The interrupt manager stores the priority of the one of the interrupt signals which initiated respective sequence of commands is currently being executed by the processor. The processors may be reconfigured programmably to operate independently in lock step or as pipelined processors. (28pp)

#### **Japan**

Publication No. JP 58058672 A (Update 198320 E)

Publication Date: 19830407

Language: JA

Priority: US 1981286426 A 19810724

⑩ 日本国特許庁 (JP)

⑪ 特許出願公開

⑫ 公開特許公報 (A)

昭58—58672

⑬ Int. Cl.<sup>3</sup>  
G 06 F 15/16  
13/00  
15/06  
H 01 L 27/04

識別記号

庁内整理番号

6619—5B  
7361—5B  
7343—5B  
8122—5F

⑭ 公開 昭和58年(1983)4月7日

発明の数 3  
審査請求 未請求

(全 58 頁)

⑮ 再構成可能集積回路

⑯ 特 願 昭57—128805

⑰ 出 願 昭57(1982)7月23日

優先権主張 ⑱ 1981年7月24日 ⑲ 米国(US)

⑳ 286425

㉑ 1981年7月24日 ㉒ 米国(US)

㉓ 286424

㉔ 1981年7月24日 ㉕ 米国(US)

㉖ 286426

⑳ 発 明 者 ロバート・エル・バドジンスキ  
アメリカ合衆国テキサス州7508

1リチャードソン・エッジウツ  
ド・ドライブ1106

㉗ 発 明 者 サテイツシュ・エム・ザット  
アメリカ合衆国テキサス州7508  
1リチャードソン・エルク・グ  
ラーヴ1304

㉘ 出 願 人 テキサス・インストルメンツ・  
インコーポレーテッド  
アメリカ合衆国テキサス州ガラ  
ス・ノース・セントラル・エク  
スプレスウェイ13500

㉙ 代 理 人 弁理士 中村稔 外4名

明 細 書

1. 発明の名称 再構成可能集積回路

2. 特許請求の範囲

(1) 複数のプロセッサと、

上記のプロセッサの全てに各々が接続される  
複数のバスと、

各々の上記プロセッサに接続されて上記プロ  
セッサの構成を選択的に変更しこれによつて上  
記プロセッサをロックステップでまたは独立に  
作動可能とする手段

を有するモノリシック基板を有する集積回路。

(2) 上記バスが複数の線を有する状況バスを有し、  
上記状況バスの上記線が同期線及び演算連結線  
を有しており、

各々の上記プロセッサが上記プロセッサの各  
各及び上記状況バスに接続される状況バスマル  
チプレクサを有し、上記それぞれの状況バスマ  
ルチプレクサが選択的且つプログラムに沿つて  
上記状況バスの選択された線に接続又は割込み  
を行い、さらに上記それぞれのプロセッサを選

択的に上記状況バスに接続し、これによつて上  
記それぞれのプロセッサが上記状況バスを通し  
て上記プロセッサのうちの隣接するものに接続  
されるようになった特許請求の範囲第1項の集  
積回路。

(3) 各々の上記プロセッサが上記それぞれのプロ  
セッサに与えられる命令を受けとる為に接続さ  
れるダイナミック型論理配列(DLA)を有し、  
上記DLAは：ANDマトリクス及びORMア  
トリクスと；上記ANDマトリクスを上記ORM  
マトリクスに接続する中間線と；上記AND及び  
ORMマトリクスの最初の1つと接続する複数の  
入力線と；上記AND及びORMマトリクスの他  
の1つと接続される複数の出力線を含み、上記  
DLAの上記AND及びORMマトリクスの少く  
とも1つが行列に配列されて、それぞれの上記  
マトリクス内の区画を規定するようになってお  
り、

上記区画は所望の論理機能を行うように配置  
された選ばれた数の能動素子を含むように選択

的に配列されており；

さらに複数の制御線が設けられ、各々の上記制御線が選択的に1つ又は2つ以上の上記区画に存在する上記能動素子を作動させたり作動を切つたりする為に接続されることで上記DLAが上記制御線の状況に応じて選択された論理機能を行うようになつており、

これによつて上記DLAが制御線の状況に従つて上記それぞれのプロセッサに選択的に与えられた命令を通訳するようになる、

特許請求の範囲第1項の集積回路。

- (4) 複数のRAMメモリモジュールを持つRAMメモリが設けられ、前記バスは複数の線を持つデータバスを有し、さらに複数のバス制御ユニットを有し、前記バス制御ユニットの各々が上記プロセッサの1つ、上記RAMメモリモジュールの1つ、及び上記データバスに接続され、上記バス制御ユニットが上記プロセッサを上記データバス又は、直接上記対応するRAMメモリモジュールに接続するようになつた特許請求

示されているかを判断し、上記割込み管理が上記プロセッサによつて現在実行中のそれぞれの命令シーケンスのいずれかを開始した上記割込み信号のうちの1つの優先順位を記憶し、対応するコンテキストスイッチ出力を与えており、

上記各々のプロセッサは上記割込み管理回路の上記コンテキストスイッチ線に接続されるスケジューラーを含み、

上記コンテキストスイッチが新しく受取つた割込み信号がより高い優先性レベルを有していることを示す時はいつでも上記スケジューラーが上記対応するプロセッサによつて現在実行される上記命令のシーケンスのとりかえを行うようになつた、

特許請求の範囲第1項の集積回路。

- (7) 上記バスは制御バスを含み、さらに上記制御バスに接続され外部ピンを有する外部割込み管理回路が設けられ、上記外部割込み管理回路は、上記対応する割込み信号のフォーマットを特定する長さに関する情報及び型に関する情報をそ

の範囲第1項の集積回路。

- (5) 各々のプロセッサの構造を変更する上記手段は、その作動によつて各々の上記プロセッサが同一の再構成可能集積回路上又は他の同様な再構成可能集積回路上の1つ又は2つ以上の上記プロセッサとロックステップ状態でまたはこれらから独立して選択的且つプログラムに沿つて作動可能である特許請求の範囲第1項の集積回路。

- (6) 上記バスが制御バスを有し、上記制御バスが命令を転送する線及び割込み信号を転送する線を含んでおり、各々の上記割込み信号が優先性及び割込み先信号を示すビットを持つて各々の命令のシーケンスを開始するようになつており、

各々の上記プロセッサは上記制御バスに接続され上記割込み信号を受取る割込み管理回路を有し、上記割込み管理回路が上記割込み信号の各々連続する1つをテストし、上記割込み管理回路を含む、上記それぞれのプロセッサが上記連続する割込み信号によつて割込み先として指

それぞれ含む割込み信号を受信及び送信する為の手段を有し上記長さの情報は、各々の割込み信号の長さを特定するようになつた特許請求の範囲第1項の集積回路。

- (8) 上記バスが複数の線を含む状況バスを有し、上記状況バスの上記線が同期及びアース連結線を有し、

上記プロセッサが上記それぞれのプロセッサ及び上記状況バスに接続されるそれぞれの状況バスマルチプレクサを有し、上記それぞれの状況バスマルチプレクサが上記状況バスの選択された線を選択的かつプログラムに沿つて接続し又は割込みを行い、更に上記プロセッサを上記状況バスに選択的に接続しこれによつて上記それぞれのプロセッサを上記状況バスを通して上記プロセッサに隣接するものに選択的且つプログラムに沿つて接続し、

各々の上記プロセッサは、ダイナミック論理配列(DLA)を持ち、上記DLAがANDマトリクス、ORマトリクス、上記ANDマトリ

クスをORマトリクスに接続する中間線とからなり、

上記AND及びORマトリクスのうちの第1のものに接続される複数の入力線と、

上記AND及びORマトリクスのうちの他のものに接続される複数の出力線とが設けられ、

上記DLAの上記AND及びORマトリクスの少なくとも1つが行列に配列されて上記それぞれのマトリクス内の区画を規定しており、

各々の上記区画が所望の論理機能を行わせる為配置された選択された数の能動素子を含むように配列されており、

複数の制御線が、その各々が1つ又は2つ以上の上記区画に存在する上記能動素子を選択的に作動させたり又は作動を切つたりすることができるように接続され、上記DLAが上記制御線の状態に応じて選択された論理機能を行うようになつており、

上記DLAが上記状況マルチプレクサに接続され、これによつて上記それぞれの状況マルチ

プレクサから、上記それぞれのプロセッサに与えられた入力及び出力が上記DLAによつてプログラムに沿つて接続されるようになつた特許請求の範囲第1項の集積回路。

(9) 複数のプロセッサと；

各々1つ又は2以上の上記プロセッサに接続された割込み管理回路とを有し、上記プロセッサの全てが上記割込み管理回路のそれぞれ1つに接続されており、

それぞれの上記外部割込み管理回路間で信号を伝達するバス手段が設けられ、

上記各々の外部割込み管理回路は、それぞれ長さに関する情報及び型に関する情報を含む割込み信号の受信及び送信を行う手段を有し、各々の上記型に関する情報が対応する割込み信号のフォーマットを特定し、各々の上記長さに関する情報が上記対応する割込み信号の長さを特定している多重プロセッサシステム。

(10) 複数のプロセッサと、

各々が上記プロセッサの全てに接続される複

数のバスとを有するモノリシック基板からなり、上記複数のバスが複数の状況線を含み、

各々の上記プロセッサがデータバスハードウェアを有しさらにプログラム可能な論理配列(PLA)を有していて、上記複数のバスの上記状況線が上記PLAの入力及び出力、また上記PLAの上記出力及び入力線に接続されている集積回路。

3 発明の詳細な説明

本発明はソフトウェアで変更が可能なプロセッサ間の接続を行う内部接続を持ち、共通メモリの全てにアクセスする多重ビットプロセッサを有し、様々な計算構成の改変も可能である再構成可能なICに関する。

VLSI技術によつて得ることのできる経済的利点をいかなる場合の主な問題点は、将来的なVLSI部品で満足のゆくレベルまで価格を低減できるほど十分な量産が可能なのはわずかしかならないということである。特に、応用例の多くの場合、プロセッサの機能にそれぞれ独自の構造を要求するので、各々の応用例に用いられるプロセッサはその目的に特に設計されることになり、必然的に高価格になる。

本発明の特に目的とすることは、1種の一般のICチップの設計を用いて多数の種々の応用例におけるプロセス上の要求を満たし、それに伴つてICを非常に経済的に製造できるようにすることである。

主としてプロセッサに加えI/O機能制御(CRTディスプレイのような)、メモリ管理、又は、特定の演算処理のための専用プロセッサを使用することは好都合である事が多い。しかしながら、上記で示した様に、個々の目的のために特別に設計されたVLSIは高価であるので、このような専用プロセッサによつて得られる利益は十分には活かし得ていない。

本発明の他の目的は、特別製ではない一般的ICを用いて特別な目的専用のプロセッサを容易に構成する手段を提供することである。

このような専用プロセッサを用いるにあつては、これらを一般的な目的のプロセッサと共に単一チップ上に集積すれば、特に好都合であることが多い。しかしながら、このような構造は更に特別な設計を必要とし、前述のように価格的不都合の問題が生じることになる。

本発明の他の目的は、容易に量産可能な一般的な目的の再構成可能ICを用いて、一般的な目的のプロセッサと1つ以上の特定目的のプロセッサ

とを単一チップ上に集積可能にすることである。

特定目的のために設計されたVLSIの他の問題点は、部品製造の歴史が長くなるほど完成部品の価格と歩留まりおよび信頼性が、「ラーニングカーブ(習熟曲線)」に沿つて良くなるという傾向があるのに対し、カスタムメイドによる特定目的の設計では、価格や信頼性に顕著な改善があらわれる程充分に長期の期間にわたり大量生産が行われることは通常はないということである。

したがつて、本発明の他の目的は、価格、歩留まりおよび信頼性がラーニングカーブに沿つて顕著に改善されるよう長期にわたり大量生産が可能な一般的な目的を有する部品を提供することである。

カスタムメイドすなわち特定目的のために設計されたVLSIは高価であるとともに、ゲートの集積度が比較的低いという問題点をさらに有する場合が多い。これは、少量で生産される部品である為にパッキングの集積度を最適とする為に多くの時間及び多くの費用を費すことは、経済的でなくなるという理由からである。

そこで、本発明の他の目的は、ゲートの高集積を保持しながらカスタムメイドの集積回路の必要を充たすことのできる集積回路を提供することである。

ゲートアレイは、非常に融通性の高いLSI又はVLSI部品を提供することができ、さらにゲートアレイは、ある種の特定の機能たとえば高速乗算器又はクロスバースイッチのような機能を有効に行うことができる。しかしながら、ゲートアレイは、プログラム可能なシステムの補助にはあまり適さないし、ゲートアレイの集積度は通常プロセッサほど高くない。

したがつて、本発明の他の目的は、ゲートアレイより高いゲートの集積度を持ち、プログラム可能なシステムの補助に一層適した再構成可能集積回路を提供することである。

比較的大規模で複雑なシステムを数学的にモデル設計したい場合、各々のプロセッサのデータ処理能力は比較的低くとも、高次の並列演算処理によつて全体的なデータ処理は非常に高くなるよう

に大規模配列でプロセッサを使用することが望ましい場合が多い。しかし、このような配列でプロセッサを使用することは、通常は特定の目的の為にシステムとして構成されるので今のところ比較的高価なものとならざるえない。

同様にして、非常に広い多重精密ワード(ここで複雑な軌道的システムが非常に長期にわたつてモデルされなくてはならない)を用いて長い連続する計算を実行することがしばしば望まれる。この場合に於ても、既製のシステムを適合させると、通常高価格又は低速となるか、又はこの両方の問題が生じることになる。

故に、本発明の他の目的は、特別なハードウェアを設計することなく、多数のこのような集積回路を結合して、安価に配列処理及び/又は多重精密処理を行うようにプロセッサを集積回路中に設けることである。

カスタムメイドVLSIのこの他の重大な問題点は、ハードウェアの設計の変更及びテストが必要とされるので設計のサイクルタイムがどうしても

も長くなることである。

そこで、本発明の他の目的は、単にソフトウェア及び／又はファームウェアを変化することによつて特定の応用例にあわせて再構成可能な乗換回路を提供することである。

多重処理システム、ダイナミック構造及びマイクロプロセッサ構造の背景を一般的に理解する為の参考文献としては以下のものがある。

「多重プロセッサ及び並列処理」(ed P. エンスロウ ジュニア 1974)；A. アプトアラ & A. メルツァ、「デジタルコンピュータ設計の目的」(1976)；C. ミード & L. コンウェイ、「VLSI システム入門」(1980)；R. クルツ、「マイクロプロセッサ及び論理設計」(1980)；G. メイヤー、「コンピュータ構造の発展」(1978)；ペアー、「多重処理システム」25 IEEE トランザクションズ・オン・コンピュータズ 1271 頁(1976)；サーパー & カルド、「協働並列プロセッサ」、7 コンピュータサーベイ 215 頁(1975)；カータシエフ & カータシエフ、

「ダイナミック構造：その問題と解決」コンピュータマガジン 7.8 年 7 月 26 頁；カータシエフ & カータシエフ、「ダイナミック構造を持つ多重コンピュータシステム」28 IEEE トランザクションズ・オン・コンピュータズ 70.4 頁(1979)；カータシエフ & カータシエフ「80 年代の為のスーパーシステム」コンピュータマガジン、1980 年 11 月号 11 頁；及びグイック「スーパーシステムに適合可能な構造」コンピュータマガジン 1980 年 11 月号 17 頁。これらは全てここに参照として示す。

使用者があまり知識を持たないような場合、特定の仕事にあわせた非常に高度なアプリケーション言語を設計することは、特に好都合である場合が多い。しかしながら、このようなアプリケーション言語がソフトウェアの形で実施される場合、速度が落ちることになるのが普通であるし、ハードウェアで実施されれば、通常、非常に高価なものとなる。

故に、本発明のこの他の目的は、あらゆる所望

のアプリケーション言語の翻訳にも適合する VLSI プロセッサを経済的に与える手段を提供することである。

多重プロセッサシステムの持つ多くの利点は、めつたに有効に利用されていない。これは 1 つには、単一プロセッサシステムより多重プロセッサにおいて最適となるプログラムを作れるようにプログラマーを訓練することは、常に困難である為である。したがつて、本発明の他の目的は、多重処理及び単一処理システムの間を簡単に構成の変更可能なシステムを提供し、プログラマーが多重プロセッサのプログラミングの利点を活かし得るようにその要求するところに徐々に慣れるようにすることである。

多重処理システムの利用に関してのこの他の問題点は、ハードウェア構造に融通性がないという問題である。多重処理システムの動作の大部分は、ハードウェアの構造によつて決定されるので、また現在のところ提案されている全てのハードウェア構造はある型の問題に対し最適でも他には最適

ではないと思われるので、真に一般的な目的を果たすシステム(汎用システム)を実現する多重プロセッサハードウェア構造はまだ 1 つもできていないと思われる。

そこで、本発明の他の目的は、再構成可能であつてこの為真に汎用多重プロセッサハードウェア構造となり得る多重プロセッサハードウェア構造を提供することである。

多くの多重プロセッサシステムに発生するこの他の問題点としては、2 つ以上のプロセッサがメモリの同じ領域にアクセスしようとする時に起こるプロセッサ間の干渉の問題である。いずれか 1 つのプロセッサによつてアクセスできるようにメモリの領域に拘束を与えることも可能であるが、これによつてプロセッサ間の干渉の問題は避けられる一方、多重プロセッサシステムによつて受ける大部分の利益を犠牲にすることになる。

したがつて、本発明の他の目的は、どのプロセッサもメモリのいずれの領域にもアクセス可能でありながら、データの完全性も保持できる多重プ



ロセツサシステムを提供することである。

マイクロプロセツサ言語と構造(アーキテクチャ)の間の、またマイクロコンピュータとミニコンピュータシステム間のエミュレーションには、現在、比較的高額な費用がかかる。もしこのエミュレーションをもつと安価にすることができれば、たとえば単一のプロセスチップで達成できるようになれば、より安価な開発及び試験が可能となり、これによつて特定目的に設計された応用システムの使用範囲が広がり、新しい技術に一番早く適合できるようになる。

本発明の他の目的は、融通性があつて効率的なエミュレーションを行うことができる単一チッププロセツサを提供することである。

故障のあつた場合をでも性能低下をおだやかにすることができるマイクロプロセツサシステムを提供することも望ましい。こうすることによつて、ハードウェアに最初に生じた重要な誤動作は、チップの機能に影響を与えなくなり、単にチップの機能をわずかに低下させるだけになるので、信頼

けとり循環させるI/O通信機能及び外部プロセツサと接続するプロセツサ間の通信機能を達成するには、これらのためにハードウェアを別々に備えたとすれば非常に多くのハードウェアを必要とする。

本発明の他の目的は、I/O通信及び中間にあるプロセツサ間通信の両方を操作できる外部インターフェースを提供することである。

本発明の他の目的は、各々のプロセツサ内でプロセツサ間通信を管理する為また外部で発生した割込みを転送する為の両方に割込みを使用できるように作られた割込み管理構造を提供することである。

プロセツサシステムの動作機能又は構造を変更する必要がある時に論理を再設計すると費用がかかり困難であるばかりでなく時間もかかる。

したがつて、本発明の他の目的は、論理を再設計するのではなくプログラミングを変更する事によつていかなるレベルにおいてもプロセツサのオペレーションを容易に変更できるようにすること

性及び歩留まりの両方が大巾に改善される。

本発明の他の目的は、ハードウェアに誤りがあつた場合、線形的な誤りとするよりむしろおだやかな機能低下をおこなすマイクロプロセツサシステムを提供することである。

非常に多数のプロセツサを持つ多重プロセツサシステムは、多重プロセツサが高価であること、これらの内部接続が困難であること及び大規模なシステム内のプロセツサ間のインターフェースの為に適当なプロトコールを特定することが非常に困難であること、などのために、今までのところその利点をわずかに利用できるのみである。

本発明の他の目的は、所要数のプロセツサを内蔵している多重プロセツサシステムの中に容易に組み込み得るプロセツサを提供することである。

ICプロセツサを多重プロセツサシステムの中に組み込み得るようにする場合、さらに生ずる問題点は、外部プロセツサとのインターフェースの為にチップ上に別のハードウェアを準備する必要があることである。外部で発生された割込みを受

である。

プロセツサのハードウェアシステムにアーキテクチャコンパイラ(architecture compiler)を提供できるようにすることは、非常に望ましいことである。このようなコンパイラは、プログラムによつて通常可能な変更に加えて、システムの有効な構成をプログラムすることを可能にする。しかしながらこのようなコンパイラを補助する為には、機能的な内部接続の再構成を選択的且つプログラムに沿つて実行できるハードウェアを備える必要がある。

そこで、本発明の他の目的は、アーキテクチャコンパイラを使用する為に必要な機能の変更が可能な構造を持つプロセツサシステムを提供することである。

本発明は、単一チップ上に4つのマイクロプログラム可能な16ビットマイクロコンピュータを備えるものである。マイクロプログラミング機能は、個々のプロセツサを含む大規模PLAにより与えられるものである。各々のプロセツサは状況

( status ) バス、データバス、及び制御バスと呼ぶ3つの各々の主要バスに接続される。それぞれのプロセッサは、プログラム可能な内部接続として動く状況バスマルチプレクサをそれぞれ介し接続される。このようにして状況バス接続の種々の構成が可能となることで、プロセッサは、(例えば、4つの16ビットプロセッサとして)単独で作動させることも、(例えば1つの64ビットプロセッサとして)ロックステップ

( lockstep ) で作動させることも、パイプラインで動作させることもできる。このように、プログラムの融通性を与える2つの主要なソースが得られる。即ち、PLA翻訳によるマイクロプログラミングの融通性であり、プログラム可能状況バス接続を使用し、各々のプロセッサによつて翻訳された命令の流れを制御することによるプロセッサ再構成の融通性である。

プロセッサ制御に於てこの融通性を利用する為、各々のチップレベルの命令は、1つまたは2つ以上の特定のプロセッサへと送られる。故に、プロ

セッサが(個々のプロセッサが同一のデータの流の中で次々と異なるオペレーションを実行する)パイプラインモードで動作することが望ましい時、各々のプロセッサはそれぞれ命令を受け、パイプラインシーケンス内のその位置に適當なオペレーションを実行する。ロックステップ処理の場合、1つのプロセッサは、(シーケンスその他の制御する)マスタープロセッサとして指定され、他のロックステップ型プロセッサは全て同時に制御される。これらの再構成の種々のモードもまた組合せ得るので例えば、1つのチップは4つの独立な16ビットプロセッサ、2つのパイプライン型32ビットプロセッサ、1つの48ビットプロセッサ(3つのロックステップ型16ビットプロセッサ)及び1つの独立16ビットプロセッサ等を含むように再構成される。制御バスが複数の関連のない命令シーケンスを運搬する為に区分され得るので、単一の制御記憶管理を用い仲裁(arbitrate)され制御バスへの仲裁が行われる多重割込み階級組織(multilevel interrupt hierarchy)が使

用される。(1つの外部割込み管理及び2つの外部状況ポート及び2つの外部データポートを含む)外部インターフェース制御によつても、2個以上のR I Cチップ上のプロセッサを一緒に前に示したような様々なモードで連結することが可能である。更に、外部インターフェース制御もまた外部メモリ、I/O装置その他へのアクセスを制御している。

どのプロセッサによつてもアクセス可能なRAMを充分量チップ上に用意する。しかしながら、各々のプロセッサもまたチップ上のRAMの主要な区域に直接アクセスしている。全てのプロセッサは、RAM内のそのプロセッサ用の優先区域に並列にそれぞれ直接アクセスされ得るが、他のRAM内の区域にもアクセスする為にはプロセッサはデータバスを使用しなくてはならないので、このようなアクセスを仲裁あるいは調整する必要がある。ハングアップ(hangup)及びデッドロック(deadlock)を防ぐ為の設計手段が存在する。第1には256の優先段階が使用され、故に優先

性の一致が起こらないようにする。第2に、全ての割込み(interrupts)が並列に送られて、確認が行われる。第3に、実行を進める為に必要とされる資源を全部は手に入れることができなかったプロセッサはいずれも、予め入手してあつた資源を全て放棄し、必要な資源が入手できるようになるまで待機する。

このような重要な構成要素に加えて、設計の仕様を完全に満たす為には多数の従来の部品も使用される。例えば各々のプロセッサは、ALU、パレルシフター、メモリマッパ、マイクロシーケンサ等を有している。

本発明による利点は、前述した本発明の目的を全て解決するものであり、他の利点も当分野に通常の知識を有するものであれば明らかであると考ええる。例えば、構造の大部分(例えば全部で4つのプロセッサ)は同じものを複製して使えばよいので、ほとんど4のファクターでR I Cチップのハードウェアの配置を考える時間が低減される。

もちろん、プロセッサは16ビットプロセッサ、

である必要はなく、変わりに32ビット、8ビットその他であつてもよい。同様にして、チップ上のプロセッサの数は、4である必要はなく3でもまた5以上でもよい。実際は、最大のプロセッサ数は、製造技術によつて単に拘束されるのであつて、より微細な加工が実用化すればたぐさの数のプロセッサをワンチップ上に作ることは非常に望ましいと予想される。このような場合、内部接続やバス及びプロトコールは、このままでよい。もちろん、プロセッサが16ビット以外の場合には、それぞれのバスの規模は、それ相当に変化させなくてはならない。

本発明は、複数のプロセッサ、それぞれが上記プロセッサの全てに接続される複数のバス及び上記のプロセッサの各々に接続され、上記プロセッサを選択的に再構成することによつて、上記プロセッサがロックステップ又は独立して動作することを可能にする手段とを有する、モノリシック基板からなる再構成可能集積回路を提供するものである。

上記プロセッサは、上記それぞれのプロセッサに与えられる命令を受けとる為に接続されるダイナミック論理配列(DLA)を有し、上記DLAは、ANDマトリクスと、ORマトリクスと、上記ANDマトリクスを上記ORマトリクスに接続する中間線と、AND及びORマトリクスの最初の1つに接続される複数の入力線と、上記AND及びORマトリクスの他の1つに接続される複数の出力線とを有し、上記DLA内の上記AND及びORマトリクスの少なくとも1つは、上記のそれぞれのマトリクス内の区画を規定する為に行列で配列され、各々の上記区画は所定数の能動素子が含まれるように選択的に配列され所望の論理機能が行われるようになっており、さらに、複数の制御線が設けられ、各々の上記制御線は、1つ又はそれ以上の上記区画内に配置される上記能動素子に接続され選択的に作動可能にしたり作動不可能にしたりして、上記DLAが、上記制御線の状況に応じて選択された論理機能を行なうようにし、これによつて上記DLAは上記それぞれのプロセッサ

また、本発明は、複数のプロセッサと、それぞれが全てのプロセッサに接続される複数のバスとを有するモノリシック基板からなり、前記バスは複数の線を持つ状況バスを含み、上記状況バスの上記線は、同期及び演算集合線(arithmetic linkage lines)を含む複数のバスとを有し、各々の上記プロセッサは、それぞれの上記プロセッサ及び上記状況バスに接続される別々の状況バスマルチプレクサを有し、上記別々の状況バスマルチプレクサは、上記状況バスの選択された線に対し選択的に且つプログラムに沿つて接続又は割込みを行い、さらに上記それぞれのプロセッサを上記状況バスに選択的に接続して、これによつて上記各々のプロセッサを上記状況バスを通し上記プロセッサの隣接する1つに選択的に且つプログラムに沿つて接続するようになつた再構成可能な集積回路を提供するものである。

さらに、本発明は、複数のプロセッサと、それぞれが全部の上記プロセッサに接続される複数のバスを有するモノリシック基板からなり、各々の

サに与えられた命令を上記制御線の状況に応じて選択的に翻訳するように構成された再構成可能な集積回路を提供するものである。

本発明はまた、複数のプロセッサと、それぞれが上記プロセッサの全てに接続される複数のバスと、RAMメモリとを有するモノリシック基板からなり、上記RAMメモリは複数のRAMメモリモジュールを有し、上記バスは複数の線を持つデータバスを有し、さらに複数のバス制御ユニットが設けられ、各々の上記バス制御ユニットは、上記プロセッサの1つ、上記RAMメモリモジュールの1つ、および上記データバスに接続され、上記バス制御ユニットは選択的に上記プロセッサを上記データバスに接続するか又は上記の対応するRAMメモリモジュールに直接接続するようになつた再構成可能な集積回路を提供するものである。本発明はまた、複数のプロセッサと、それぞれが上記プロセッサの全てに接続される複数のバスとを有するモノリシック基板からなり、それぞれの上記プロセッサを再構成する手段が設けられ、

これによつて各々のプロセッサは同一の再構成可能集積回路上の1つ又は2つ以上の他のプロセッサといつしよにロックステップ形式で又は独立して、選択的且つプログラムに沿つて作動できるようになつた再構成可能な集積回路を提供するものである。

さらに、本発明は、複数のプロセッサと、それぞれが上記プロセッサの全てに接続される複数のバスを有するモノリシック基板からなり、上記バスは制御バスを含み、この制御バスは命令を送送する線と割り込み信号を送送する線とを有し、各々の上記割り込み信号は、優先性及び上記プロセッサの中の割り込み先プロセッサ (interruptee processor) を示すビットを有し、この上記割り込み信号の1つが上記命令のシーケンスを開始させるようになつており、各々の上記プロセッサは、上記制御バスに接続し割り込み信号を受取る割り込み管理回路 (interrupt manager) を有し、この割り込み管理回路が上記割り込み信号の連続する1つをそれぞれテストし、上記割り込み管理回路を持つ上記

それぞれのプロセッサが上記連続する割り込み信号によつて割合て先として指定されたか否かを決定するようになつており、上記割り込み管理回路は、現在、上記プロセッサで実行される別々の命令シーケンスのいずれかを開始させた上記割り込み信号の1つの優先性を記憶し、上記割り込み管理回路は、各々の上記プロセッサにアドレスされた上記割り込み信号の各々連続する1つの優先性と、現在、上記プロセッサで実行されている別々の命令シーケンスのいずれかを開始させた上記それぞれの割り込み信号の優先性を比較し、それに相当するコンテキストスイッチ出力 (context switch output) を与えるようになつており、各々の上記別々のプロセッサは上記割り込み管理回路の上記コンテキストスイッチ線に接続されるスケデューラ (scheduler) を有し、上記スケデューラは、上記コンテキストスイッチが新しく受取つた割り込み信号がより高い優先レベルであると示す時にいつでも、現在上記対応するプロセッサで実行されている上記命令シーケンスの交換を行なうようになつた再

構成可能な集積回路を提供するものである。

以下、本発明に関し実施例を用い図を参照しながら詳細に説明する。

第1図は、再構成可能IC (以下はRICと略す) のブロック図を示す。PR0からPR3までの4つのプロセッサは単一チップ上に形成されており、全てのプロセッサ間を接続する為、3つのバス14、52及び56が設けられている。各々のバスに対応して外部インターフェース76、74と75、及び72と73が設けられ、またオンチップRAMメモリ66及び制御記憶回路38 (第2図) も形成される。より詳細な配置図は、第2図で示されている。第2図は、1つのプロセッサPR3の全体と、それぞれの型の外部インターフェースのうち1つ及びバス線とオンチップメモリの一部を含む再構成可能ICの部分的フロアプランを示す。各々のプロセッサの割り込み管理回路12は、制御バス14上に現われる割り込みを常に監視している。プロセッサPR3にアドレスされた割り込みだけが、PR3内の割り込み管理回路

12によつて認識される。このような割り込み信号はPR3内のスケデューラ16によつて連続して比較され、これらの信号の優先レベルがプロセッサPR3で現在実行中の命令シーケンスの優先性より高いかを調べる。これが高い場合であれば、新しい割り込みが導入した命令の流れ (このような命令の流れは「プロセス」と呼ばれる) の実行の開始が必要である。スケデューラ16は、対応する出力を制御DLA22に与え、こればROM命令レジスタ18又はRAM命令レジスタ20を作動させるので制御DLA22が「プロセス」の中に含まれる命令のシーケンスを受けとり始める。制御DLA22は、以下に示すようにその構造上の改良によつてパッキング集積度が非常に向上されている点を除けばPLAと同様に機能する。したがつて、制御DLA22は、その内部のANDマトリクスにミンターム (minterms) を発生し、次にORマトリクスがこれらのミンタームを次に選択された論理和出力に変換する。制御DLA22と隣接するのは、フィードバックブロック

24である。これを通つていくつかの制御DLA 22の出力はフィードバックされ、DLA 22の入力へと接続される。この意味で一定の状況による機械機能がDLA 22に与えられるので、DLA 22は例えばある高レベルの命令を低レベルの命令のシーケンスに翻訳することが可能である。

DLA 22は、デコード区域25及び26に接続され、これによつてDLA 22の出力はALU 28、パレルシフタ30、レジスタファイル32、メモリマップパ34等に対するハードウェア命令として選択される。DLA 22の出力は、またマイクロシーケンサ36に接続され、これによつて制御バス14及び制御記憶回路38に対するアクセスの制御を行う。マイクロシーケンサ36は制御バス14上に適当な信号を送ることが可能で制御記憶アドレスレジスタ40を通つて制御記憶回路38にアクセスしている。制御記憶回路38から呼び出されたデータは、制御記憶データレジスタ42を介し制御バス14に出力され、更に、もレジスタ18が割込み管理回路12によつて作

動されている場合には、ROM命令レジスタ18内に入力され、制御DLA 22のAND区域に対する入力として接続される。メモリマップパ34、レジスタファイル32、パレルシフタ30、ALU 28、シフトレジスタ44及びフラグレジスタ46のオペレーションは、マイクロプロセッサ技術としては既知の従来の主要オペレーションに従い行われる。

エンド・アラウンド・ループ54を含む状況バス52もチップ上に形成される。状況バスは、状況マルチプレクサ45及び50を介しそれぞれのプロセッサに接続される。各々のこのようなバスの状況マルチプレクサはプログラム可能な相互接続として作動する。状況バス52及び54の線は、パイプラインモード中の隣接するプロセッサ間に同期を与える為に又はロックステップモードでの隣接するプロセッサの間に完全な直結を作り出す為に使用される。単独モードでは、このような連結はまったく必要ない。1対の状況マルチプレクサ45及び50はプログラムされて、新しいモー

ドが入ってくるタイミングごとに適当な状況バスの相互接続が行われる。

プロセッサPR3は、メモリマップパ34を介してRAMメモリ66にアクセスしている。メモリマップパ34からの線は、状況バス52を54と交叉しており、バス制御ユニット58でデータバス56とインターフェースしている。バス制御ユニット58は選択的にプログラムに沿つて作動するので、メモリマップパ34からの出力は、プロセッサPR3が好ましくはアクセスするRAMメモリモジュール60に、データレジスタ62及びアドレスレジスタ64を通つて直接接続するか又はデータバス56を通つてチップ上又はチップ外のメモリのどこか他の領域に接続される。RAMメモリ66の他の3つのモジュール(図示せず)のうちの1つにアクセスする必要がある場合、このアクセスはデータバス56を介し、他の3つのメモリスケデューラユニット68のうち適当な1つと接続される。各々のメモリスケデューラユニット68は、先に来たものから先にサービスを行

う方法に従つてメモリアccessの要求をスケデューリングし、制御レジスタ70を通つて相当するメモリモジュールへアクセスするよう制御する。更に、データバス56は外部データポート72及び73(第1図参照)に接続され、これを通して各々のプロセッサは、チップ外メモリにアクセス可能となつている。

同様にして、状況バス52及び54は、外部状況ポート74及び75に接続されるので、チップ外プロセッサは、多重チップ間のロックステップ及びパイプラインのオペレーションで同期するようになり、制御バス14が外部割込み管理回路76に接続されるので、チップ外プロセッサとの間で命令の送信受信を行うことができる。

制御記憶回路38は、それぞれの制御記憶モジュール制御器96によつて制御されるモジュール78に分割されている。RAMメモリ66も同様にモジュール60に分割されている。最後に複数のパッド接続領域80が外部との接続の為にチップ周辺に設けられている。また84ピンパッケージ

を使用することが好ましい。

前述した内容は、再構成可能集積回路の機能及び構造を示す概略的説明である。チップに関するさらに詳しい説明はいくつかの実施可能な応用例としての提案を行いながらこれから説明する。

これからさらに詳しく説明する再構成可能な集積回路の第1の特徴は制御バスである。制御バスの重要部分の役目は、制御DLA22によつて行われ、このDLAの構造及びオペレーションを第1に説明する。各々のプロセッサに用いられるプログラム可能な論理配列PLAは大規模なものであることを必要とする。本実施例では、 $40 \times 400 \times 120$ のPLAを必要とする。現在の製造技術では、この程度の複雑なPLAは、非経済的な程に広い面積である10,000平方ミルのオーダの領域を占めることになる。しかしながら、プロセス技術の改良が必要な領域を非常に小さくしている。更に、上記のような従来のPLAのかわりにDLA(ダイナミック論理配列)を用いることでかなりのサイズに関する改良が可能となつた。

機能ではなく多重機能の為にPLAの入力及び出力を使用可能にすることができる。DLAは、単一機能を実行する回路を作動させ一方この他の機能を実行する回路をオフにする方法でPLAの部分を絶縁することによつてこれを行つている。

DLA22内の能動部分の選択は、2つのデコード25及び26によつて行われる。これらの2つのデコードによつてDLAの制御線が作動されることに従つて、DLAが作動し、所定の論理機能を形成する。もちろん、DLAのいくつかの部分はこれらの制御線によつて制御されていない。この考え方は、PLAにおけるNORゲート仕様及びNANDゲート仕様の両方に適用可能である。

第3図に示すように、NMOS技術を用いてのPLAのNORゲート仕様においては、ORマトリクス内にゲートを形成するトランジスタと共にANDマトリクス内にゲートを作るトランジスタは、共通して接地されるソースを全て有している。特定の機能を作り出すトランジスタのソースがフロートされている場合、回路は、これらのトラン

ジスタが多い用途には、マスクプログラムによる

DLAを用いることが好ましいが、初期の開発段階ではフィールドプログラム可能な(又は電氣的に変更可能でさえある)DLAを使用すると、需要に対する融通性を拡げることができる。集積度に関する厳格な最高の限界は所望のミントーム又はマクスターム(minterm又はmaxterm)の結果にかかっているが典型的なPLAはトランジスタ集積度の10%を有しているので、DLAの使用により4倍の改良がミントームを「折りたたむ」ことによつて容易に可能となる。ダイナミック論理配列DLAは、領域をもつと有効に利用する為に改良されたPLAとして簡単に考えることができる。そのかわりにDLAを回路レベルで再構成可能にしたPLAとして考えることもできる。

DLAによつて多重機能をPLAで実施できるようになるが、一度に1つの機能しか利用できなくなるという拘束が与えられてしまう。DLAは2から5倍の率で領域の有効利用が改善される。(マルチプレクサと共働する)DLAはまた単一の

ジスタが取り除かれたかのように働くのでその機能は働かなくなる。NORゲートDLAの背景にある基本的思想は、機能を行う為の全てのトランジスタを制御線に接続し、機能を動かせる時に制御線は接地され、機能を働かせない時にフロートにすることである。DLAで多重機能を作り出す為には、各々の機能を作り出すトランジスタのソースを、その機能に対応する別々の制御線に接続する。第3図は、典型的なNORゲートDLAを示している。通常のPLA回路は、実線で示され、DLAを作る為の追加的回路は、点線で示されている。DLAは破線によつて示されて区分されている。種々の区分は、P1からR8までの表示が付されている。制御線C1が論理1である場合、区分P1及びP5に於る回路が作動され、出力E及びFは、入力A、 $\bar{B}$ 及びBの関数となる。制御線C1が0レベルである場合、E及びFの出力は、いずれの入力の関数としても制御されることもない。追加の回路もまた出力E及びFを入力A、 $\bar{B}$ 及びBの関数として制御される他の

機能を作り出す為にP3及びP7の区域に追加することができる。

同様にして、第3図の制御線C2は区分P4及びP8に於る回路を作動又は非作動状態にすることができ、追加の出力機能を与える追加の回路を区分P2及びP6内に追加することができる。

この回路は、あらゆる数の区分に対しても一般化することが可能であつて、この区分は様々なサイズが可能である。またANDマトリクスは、ORマトリクスからは分離して制御可能である。

第3図は、スタティックゲートPLAに基づいて改良が行われたDLAを示しているが、明らかに、上記の説明はダイナミックゲートPLAに基づくDLAにも応用することはできる。ダイナミックゲートDLAを作る為に必要な追加の周辺制御回路は、特にPLAが大規模である場合でも非常にわずかなチップ領域を増加するだけで足りる。

第4図はNANDゲートDLAを示す。多重化された機能の選択された1つを作り出す為にPLA

を区分に分けるといふ考え方はそのままである。変わつたのは所望の機能を作り出す為のゲートを絶縁する方法に関してである。NORゲートを作り出す為に、(共通ソースの)電圧を制御する方法が使用される。NANDゲートDLAでは、電圧制御されたNORゲートDLAに対し電流はデュアルで流れる。NANDゲートDLAにおいては、所望の機能を作り出すトランジスタ以外の全てのトランジスタは制御線によつて制御される電流バイパスによつてう回(シャント)される。所望の機能を作り出すトランジスタのみが上記のようにう回されることはない。マクスタームよりむしろミンタームの結果を発生するDLAを構成することも可能であるが領域の経済性は同じままである。

翻訳の為にDLAを使用することによつて、各々のプロセッサの翻訳システムは、DLAのオペレーションのモードを選択することによつて容易に再構成可能となる。より高価であるフィールドプログラム可能構造も使用者の開発にとり有用で

あるが、DLA22はマスクプログラムによるものを使用することが好ましい。

このように、制御DLA22を使用することで、各々のプロセッサ内で必要とされる複雑なマイクロプログラミング機能を得ることができる。マクロ命令を入力としてDLAに与えているバスについて以下に説明する。

DLA22は、また単にリアルタイム入力をDLAの入力線の1つに接続し、その入力とDLAの適当な出力をAND接続することによつてリアルタイム機能を提供する為に使用されている。

制御記憶回路及び制御バスの構造及び使用する割込み制御システムにつき以下説明する。

ある実施例に於ては、中央制御記憶回路は、4つの全てのプロセッサにアクセス可能で共用が可能である。ここで示す好ましい選択例では、制御記憶回路38のどの部分にも各々のプロセッサがアクセス可能なままにしておきながら、プロセッサの中の制御記憶回路38は、プロセッサの中のデータ記憶部と同様の配置で、配置されている。

4つのマイクロシーケンサ36は、並列アクセスが可能でないが、単に低い平均アクセスタイムは可能である。もちろん、マイクロシーケンサ36は(バス制御ユニット38と同様に)並列アクセスが可能なのにも構成されるが、これにみあうだけの利点は得られずに、追加の回路によつて複雑性は増すことになる。中央共有可能制御回路は、以下のような利点を与えている。多重プロセッサで使用されるコードは複製されないで、メモリ領域をより有効に利用することができる。各々のプロセッサに与えられる制御記憶量を要求に対して、より良い具合に合わせることが可能になる。メモリを中央に集中できるのでフィールドプログラミングが実用的である。もちろん、中央制御記憶回路にアクセスする単一のチャネルがシステムの障害となるのを避ける為に、制御記憶回路に記憶される命令のレベルを充分に高くして、各々のプロセッサが制御記憶回路から受けとつた各々の単一命令を、実行する為に平均して4又はそれ以上の完全クロックサイクルを要するようにしな

くではない。現在のマイクロプロセッサのソフトウェア構造に於ては、1つのアッセンブリ言語命令には典型的に平均して5ないし10サイクルを要するので、命令のレベルに関する上記の拘束は、容易に満たされる。

マイクロシーケンサ36は、ROMメモリ内に記憶されるマイクロ命令を実行するシーケンスを制御する為のアドレスシーケンサである。シーケンス的なアクセス機能に加えて、このマイクロシーケンサに14ビットの範囲又は16Kワードの範囲内であらゆるマイクロ命令を条件付きで分岐している。またマイクロサブルーチン回帰連結及びループ機能を与えているラスト・イン・アンド・ファースト・アウト形式のスタックをこのシーケンサが提供している。ここでは、4から8のレベルのマイクロサブルーチンがある。アドレス範囲内のマイクロ命令ループカウンタも有している。マイクロシーケンサ36の実施例についてのこの他の詳しい説明は、当分野に通常の知識を有する者には明らかであると考え、さらに詳細を第47

図に示すにとどめる。

中央制御記憶回路構造は、中央共有制御バスを含んでいる。中央制御バス14は4つのプロセッサPR0からPR3の中の制御記憶回路38の使用の仲裁を行い、プロセッサからのアドレスを転送し、制御記憶回路38からのマイクロ命令を複数のプロセッサに転送している。制御バス14は、チップの長さと同じ位の長さで設けられた73本の線を有している。故に、バスは高価な資源であつてバスの帯域をフルに使用できるようにすることが望ましい。即ち、制御バスの帯域は制御記憶回路38のスロットップ及びプロセッサのマイクロ記憶アクセス率とほぼ一致する。

本実施例に従つた制御バス14の構造が第5図に示される。制御バス14は制御データバス82、アドレスバス84、割込バス(destination bus)86、割込みバス88及びデタルーチンバス90を有している。制御データバス82は、制御記憶回路38からとつてきたマイクロ命令をプロセッサPR0からPR3までに転送している。このバスは40本分の線の広さである。アドレスバス84は、マイクロシーケンサ36によつてそれぞれのプロセッサ内で発生されたアドレスを制御記憶回路38に送り、マイクロ命令をとりだしている。以下に説明するように、アドレスバス84もまた割込み期間中は優先性の情報を運んでいる。このバスは14本分の線の広さであつて、制御記憶回路38に16Kワードのアドレス領域を確保している。割込バス86はプロセッサが制御記憶回路38にアクセスする時にいつでも使用され、1つまたそれ以上のプロセッサPR0-PR3及び外部割込み管理回路76のいずれかで現在アクセスされている制御記憶回路38内のア

ドレスに記憶されるデータを受けとるかを知らせる。割込バス86は、割込み期間中に使用されるプロセッサPR0-PR3及び割込み管理76のうちのいずれが現在の割込みを受け取るか(即ち割込み先interruptee)を確定している。割込バス86は線5本分の広さである。割込みバス88は割込みのソースを転送するのに使用される。このバスは、割込み先プロセッサに対し割込みが手元にある中で最も高い優先性のタスクであるかどうかを知らせる為にも(即ち割込み先プロセッサが割込みを受入れるかを知らせる為に)使用される。この型の応答では、パイプライン型又はロックステップ型オペレーションを必要としてプロセスを迅速に設定(又は延期)する必要がある。割込みバス88は5本分の線の広さである。デタルーチンバス90は制御記憶回路92によつて制御され、PR0からPR3のうちの1つ又は2つ以上のどのプロセッサが現在制御バス82で転送中のマイクロ命令を受けとるのを示す為に使用される。デタルーチンバス90は5本分の線の



広さである。線DR0 - DR3は、ROMからとり出されたワードの割当て通りに又は割込みに応じてPR0からPR3のうちの対応するプロセッサを指定する為に使用される。線DR4はROMからとり出されたワードの割当て又は割込みに応じて外部割込み管理回路を指定するために使用される。

第6図に示されるように、制御記憶管理回路92は1つの中央制御器94及び4つの制御記憶モジュール制御器96から組織される。第7図は、制御記憶管理回路92のオペレーションを示す流れ図である。制御記憶管理回路92が作動される時、この回路は、バス86、84からそれぞれ割当て及び制御アドレスの情報を受取る。次に回路は、アドレスバス84で受けとつたアドレスのうちの上から2つの最高位ビットの示す内容に従つて制御記憶回路38内の4つの制御記憶モジュール78のうち1つを選択する。アドレスは更に適当なモジュール制御回路96に送られる。第7図の流れ図に示す上述の機能は、中央制御回路94

位をあげるかのうちいずれかを実行する。あるプロセッサが制御バスを使用してバス作動線が循環してそのサイクルの間にこのプロセッサに戻ってくる場合（即ち他のプロセッサがいずれも制御バスを使用する必要のない時）バスを利用してあるプロセッサは信号を受けとり、制御バスの同期を適正に保つ為、信号を改めて発生する。制御バスのアクセスにはラウンドロビン（round-robin）即ち優先性を決めないで）仲裁が行われる。従来の型式と同様に、アドレス準備線100及びデータ準備線102を使用してアクティブなプロセッサ間及び／又は制御記憶管理回路92間での通信に同期を行う。この他のデータ準備線及びアドレス準備線を使用して割込みの転送期間中のアドレスとデータとの間の伝達を行う。

最後に、制御バス14は、花輪状に連結する、割込み準備線104も有している。花輪のように連結することによつて割込みを送るうとしているプロセッサ間でラウンドロビン型の仲裁を行つている。プロセッサが制御バス14を支配して、た

によつて実行され、残る機能は、適当なモジュール制御回路96によつて実行される。モジュール制御96はここで要求されたアクセスの為のアドレス及び割当て先を持ち行列にしてFIFOのオーダーで要求された制御ワードを読み出す。

制御バス14による仲裁は、プロセッサPR0からPR3の間を花輪状に結ぶバス作動線98（第5図参照）を用いて行われる。あるプロセッサが制御バス14を支配するとき、このプロセッサはフルの1サイクルのみの間制御バスを使用する。（アドレスを送る為の第1フェイズ、及び連続する必要はないがデータを受けとる第2フェイズ）。第1フェイズの間、バス14を制御するプロセッサはバス作動線98の電位を低い状況にしておく。アドレスが送られた後、このプロセッサは、バス作動線98の電位を上げる。線は花輪状に連結しているので、バス作動回路98の高論理状態は1つのプロセッサだけによりただちに受けとられて、このプロセッサは制御メモリへアクセスするか又はバス作動線98の花環状の連結の次の連結の電

だちに割込みを発生する時、プロセッサは、花輪状に連結する割込み準備線104にゼロまで引き下げる信号を送る。割込みを送るプロセッサは制御アドレスバス84を支配する前に割込みバス88を引き継ぐことができる。割込み元プロセッサ（interrupting processor）が割当てバス86を支配する時、発生した割込みの割当て先である外部割込み管理回路76上のプロセッサに対応する割当てバス86内の線の電位を上げる。割込みを発生した後、発生元のプロセッサは、自分が割込みの発生元であることを割込みバス88上に示し知らせる。

割込みが開始された後、他の割込みは少なくとも2バスサイクルの間禁止される：1サイクルは割込みを送る為で1サイクルは、受け取り側プロセッサ（割込み先 Interruptee）からの応答を発生元プロセッサ（割込み元 Interruptor）が受けとる為である。割込みをパツプアすることだけが必要な場合、割込み先の割込み管理回路は割込みのプロセスは可能で1から4の追加バスサイクルが

過ぎるまで他の割込みを受けとる用意ができてい  
る。割込みの受け取り側がロックステップ又はバ  
イブラインモードで連結するようになった場合、  
即ちコンテキストスイッチが必要とされる場合、  
割込みバス88は、必要なコンテキストスイッチ  
を実行する為に使われるタイミングの間ブロック  
される。コンテキストスイッチの長さは、割込ま  
れたプロセッサ内にどれほどのコンテキストが（  
即ちレジスタの内容、ALU状況等）保持されて  
いるかに左右される。以下で説明する通り、いく  
つかのコンテキストスイッチはプロセッサのコン  
テキストを本質的に完全に変更することが必要で  
ある一方、他のスイッチは最小の変更のみを要す  
る。最も短いコンテキストスイッチでも通常3又  
はそれ以上のバスサイクルを必要とする。この遅  
延が割込み信号の帯域幅を限定するが、割込みが  
フルに有効な帯域幅を占めるとは予想されないの  
でこの方法は適していると思える。この遅延は長  
い実行を通しての割込み信号の運行の平均したレ  
ートを落とさずに突然の割込みの発生をスムーズ

ROMアドレスにする翻訳を行つていて更にこれ  
らのROMアドレスはマイクロシーケンサを介し  
てアクセスされている。また、RAM命令は、  
DLA22によつて直接デコードすることができ  
る。ROM内に記憶される制御ワードはプロセッ  
サ仕様フィールドを有していない。しかし、制御  
ワードは、唯一固定された定型部分がオPCODE  
（op code）である短いビットのストリング（好  
ましくは32ビット）である。ROMから呼びだ  
された制御ワードがプロセッサのDLA22に回  
帰され、制御ワードと共にRAMから受取つた命  
令によつて特定されるあらゆる定数及びオペラ  
ンドがDLA22によつて通訳される。DLAは、  
OPコードと同時に例えばオペランドフィールド、  
定数フィールド、マイクロシーケンサ命令、メモ  
リインターフェース命令、状況バス命令、（命令  
が割込みである場合の）割込め及び優先性データ、  
割込み管理回路の制御の為の命令、スケジューラ、  
パレルンフタ等であるOPコード及び制御ワード  
に従つて残りの制御ワードフィールドを通訳する。

にする目的を果たしている。

割込みは、それぞれの命令のシーケンス（即ち  
各々の「プロセス」）を開始する為に使用される。  
上記で示した様に、割込みは優先レベルを特定し、  
データルーチンバス90内を移動しどのプロセッ  
サがアドレスされるかを特定する4ビットコード  
を有している。もし割込みによつてアドレスされ  
る全てのプロセッサが有効な状態にあるならば、  
（即ち、これより高い優先性のタスクは手元にな  
い場合）プロセス内の命令シーケンスの実行が開  
始される。もし、実行すべき命令が制御記憶回路  
38に存在している場合、この命令は適当なマイ  
クロシーケンサ36によつて次々と読出される。  
実行すべき命令がRAMメモリ66内にある場合、  
この命令は、適当なメモリスケデューラユニット  
68及びバス制御ユニット58によつてRAMメ  
モリから次々と読出される。各々の影響をうける  
プロセッサ内のメモリマッパ34はこれらの命  
令をRAM命令レジスタ20を通しDLA22に  
転送する。DLA22はこれらのRAM命令を

命令ワードの適当な部分が更にデコードされ、デ  
コード25及び26に制御されるDLAによつて  
適当なハードウェアに直接接続される。

もちろん、このマイクロプログラミング通訳構  
造では、ハードウェアでの実行以前にこれ以上の  
ステージを持つ通訳機構を用いることも可能であ  
る。例えば、制御記憶回路38から読出された命  
令は通訳され、制御記憶回路38又はRAM66  
から読みだされた他の連続する命令の実行を要求  
するようになる。更に、チップ外メモリも命令の  
実行に使用される。例えば、アプリケーション言  
語から成る単一命令が非常に長々しいアプルーチ  
ンを示している場合、制御記憶回路38内の相当  
する制御ワードは、通訳されて、RAMメモリ  
66の特定ブロックにあるサブルーチンをロード  
し、引き続きそのサブルーチンの命令を実行せよ  
という命令が出る。もちろん、チップ外の記憶容  
量に記憶される命令はとりだされ以下にさらに詳  
しくのべるように割込み管理回路76及び／又は  
外部データポート72及び73を通つて実行され

る。

割込みプロトコールに関するこれ以上の説明は以下で示す。以下で示すRICを異るモードのオペレーションに変更する再構成機構は制御バスのオペレーションに関連づけて説明する。この点に於て、プロセッサPR0-PR3の各々の構造についてもさらに詳しく説明する。

第2図からわかる通り、各々のプロセッサは、DLA22及び共働デコーダ25及び26に加えて、フィードバック線24及び命令レジスタ18及び20、割込み管理回路12、スケデューラ16、メモリマッパ34、スタック/レジスタファイル32、バレル(barrel)シフタ30、ALU28、シフトレジスタ44及びフラグレジスタ46を有している。後半の構成部は従来通りであるが、便宜上さらに明らかにする為に詳細に説明する。例えば、第9図は、ALU28をさらに詳細に示している。2本のバス108及び110は、プロセッサ内の他のブロック例えばバレルシフタ30からの入力として提供されている。この

供している。(DLA22内の)フラグ発生論理134は、フラグレジスタ46に接続され、出力ラッチ136はシフトレジスタ44に接続される。次にシフトレジスタ44はバスA及びB(108及び110)を通つてレジスタファイル32に接続されるか又はメモリマッパ32を通してバス制御ユニット58に接続されるので、ALUの出力は、データバス56内を転送されるか又は、将来のオペレーションで使用する為にプロセッサ内に記憶される。

第10図は、ALU28内の機能ブロック116、118又は122の1つの一部分の部品レベルの構造を示す。第11図は、ALU28内の桁上げ連鎖ブロック120の一部分の構造を示す。マイクロプロセッサの構造に関するこれ以上の参考は例えばオズボーンアンドアソシエートに示され、これは参考としてここにいつしよに示す。

第12図は、データバスの概略図である。入力信号は、バス108及び110から入力され、これらのバスにはそれぞれ入力ラッチ130及び

バスはそれぞれの入力ラッチ112及び114を介して、P(伝播)機能ブロック116、K(連断)機能ブロック118、C(桁上げ連鎖)ブロック120及びR(結果)機能ブロック122に連続して接続される。伝播機能ブロック116は、P制御線124によつて制御され、連断ブロック(kill block)はK制御線126によつて制御され、結果ブロックはR制御線128によつて制御される。これらの制御線はDLA22からALU28までを接続している。更に、桁上げ入力(carry in)線130及び桁上げ出力(carry out)線132が提供されていて、Cブロック120を状況マルチプレクサ48及び50に接続する。(状況マルチプレクサ48及び50のオペレーションは、状況バスプロトコールに関連して以下でさらに詳しく説明する。)Rブロック122の出力はデータバスであつて、フラグ発生論理134を通つて出力ラッチ136に接続されている。フラグ発生論理134は状況とエラーの情報を計算しDLAを介しプログラム状況ワードを提

140が接続されている。これらのラッチはバレルシフタ30の入力と接続する。バレルシフタ30は、制御線144及びパラメータ入力146によつて制御されている。これらはまた両方ともDLA22から提供されている。パラメータ入力146は、例えばシフトカウンタの為、及び抽出の限界(extraction boundaries)の為の値を与えている。バレルシフタ30はバスA及びバスBに出力を与えている。ALUは、桁上げ、オーバーフロー、負数及び/又はゼロといったような演算状況信号を発生している。ALUは、整数のオーバーフロー、十進数の桁上げ等といった他の割込み信号も発生している。このような信号は、DLAに送られてDLAは、これらの信号に従つて論理オペレーションを実行し信号を発生している。この信号は、フラグレジスタ内でビットをセットする。この機構によつて状況セット信号を発生させ、例えばコンピュータ構造の中を循環させる。

第13図は、バレルシフタ30によつて形成さ

れるシフトオペレーションで使用する用語を示している。従来、左シフトは、最上位ビットに向かってシフトすることを示す為に使用される。即ち、最上位ビットは左シフトに関しては先頭ビットである。同様にして最下位ビットは、左シフトに関する最後尾ビットであつて右シフトに関しては、先頭ビットである。第14a、b及びc図は、シフトオペレーションのこの他のいくつかの例を示している。第14a図は、最後尾ビットにゼロが入り、先導ビットが欠けている左シフトを示している。

第14b図は左循環オペレーションを示しており、入力ラッチA 138の最上位ビットは、入力ラッチBの最下位ビットにマッピングされ、入力ラッチB 140の最上位ビットは入力ラッチA 138の最下位ビットにマッピングされ、他のビットはそれに従つてシフトされている。第14b図は、1ビットの循環オペレーションを示しているが、循環オペレーションは、パレルシフタ内の1クロックサイクル期間中に16ビットまで右又

は左に動きうる。

第14c図は、シフト及び連結オペレーションを示しており、ここで入力ラッチ138は右にシフトし、連結入力線(link-in line)152が最後尾ビットの新しい内容を与えている。先頭ビットの内容は、連結出力線(link-out line)154を通り出力される。本実施例に於て、連結入力152及び連結出力線154は、入力ラッチB 140から接続されているので多重ビットシフト及び連結のオペレーションが実行されるが個別の連結入力及び連結出力線を代わるがわるに用いることもできる。

第15図は、パレルシフタ30によつて実行される抽出(extraction)オペレーションを示している。ここに示すオペレーションでは、パレルシフタ30は命令を受けて入力3-11ビットを抽出している。これらのビットは出力ラッチ150に先頭ビットとして接続される。このような機能をパレルシフター30に実行させる為に必要とされる回路は既知のように例えば先に参照と

して示したミード及びコンウェイの「VLSIシステム入門」に示されている。

メモリマッパー34はプロセッサ構造としてはよく知られる機能を形成する単なるありふれた構造である。メモリマッパー34によつて受けとられる仮想的(ヴァーチャル)アドレスは共働するメモリオペレーションによつて16の記憶された仮想アドレスと並列に比較が行われ、もし受取つた仮想的アドレスが記憶されていた仮想的アドレスの一つと整合する場合、これに相当する地域的地址が使用され所望のワードが地域メモリから呼び出される。整合するものがない場合、仮想的アドレスは(ページテーブル索引のような)テーブル索引によつて翻訳され実際のアドレスを決定し、レジスタファイルの1つに随意にロード可能となる。この意味で、メモリマッパー34は内部的に記憶されるメモリユニットと外部的に記憶されるメモリユニットを区別しているといえる。故にメモリマッパー34はテーブル検索をこま切れにする(hashing)ことができるように構成さ

れ、ハッシュテーブル内の位置を選択する為に使用されたハッシュ機能はファームウェアによるプログラムが可能である。ハッシュテーブルはマルチウェイセット共働メモリとして作られている。ハッシュ機能の出力は並行して検索が行われる多重位置を指示する。所望のアドレスがハッシュテーブルで多重位置に於て行われた比較と整合しない場合所望のアドレスは(設計上)テーブルに存在しないということなのでこれ以上の検索は必要なくなる。故にこのマッピング機能は様々なサイズのメモリのマッピングを行う。ハッシュ機能が記憶容量を補助する為に使用される場合、ハッシュテーブルエントリーが数10バイトのメモリユニットを指示する。ハッシュ機能が仮想的メモリ翻訳ルックアサイドバッファを補助する場合、メモリユニットは典型的に512バイトから2,048バイトまでになる。メモリマッパーに適當な構造をもつ物としては、ナショナルセミコンダクタ社のチップ番号16082がある。

割込み管理回路12は制御バス14を通つて転

送されてくる全ての割込み信号を直接受取っている。(割込みプロトコールは以下でさらに詳しく説明する。)割込み管理回路は、各々の割込みごとに適当な割当てビットをテストし、その割込み管理のプロセッサが割込み先であるか否かを調べアドレスが適正である場合には、新しい割込みの優先性と現在実行されているプロセスの優先性を比較する。新しいプロセスの方が高い優先性を持つ場合、割込み管理は割込みを受け入れ、さもなければ割込み管理回路12は、ワイヤーANDED線の電位を引き下げて割込みを拒絶する。

スケデューラ16は256ビットシフトレジスタ内の優先性によつて割込みをパツファする。プロセスが有効に行われている時、スケデューラ16はシフトレジスタ全体をスキャンし、次に高い優先性を持つプロセスを見つける。現在のプロセスが完了又は時間切れの場合、スケデューラ16は次に高い優先性のプロセスの優先性を使つてこのプロセスのコンテキストを示すポイントを含むテーブルにアクセスしている。適当なコンテ

スタック/レジスタファイル32は、2本のバス108及び110を十分に利用できるようにする為にデュアルポートのレジスタファイルでなくてはならない。さもなければレジスタファイル32は従来と全く同じである。

データバス56の組織は、RAMメモリ66、バス制御ユニット58及びメモリスケデューラユニット68のオペレーション及び構造といつしよに以下で説明する。

第16図は、データバス56の構造を示している。データバス56は16本のアドレス線156、16本のデータ線158、ラウンドロビン仲裁線160、4本のソース線162、モジュールビジョ線164、ソース状況線166及びBCUモード線167を有している。別々のアドレス線156及びデータ線158によつてデータアクセスのオペレーションが単一バスサイクルで実行可能となつている。この場合チップ上のRAMメモリ66がアクセスされるので少なくともメモリアクセス速度はこれを可能にしている。こうする代わりに、

クストは、スタック/レジスタファイル32から、RAMメモリ66から又は外部メモリから再び呼び出される。

第18図は、上記で示したデータバスの総体的な全体図を示す。バスA及びB(108及び110)は、主要な構成部であるデータバス、即ちスタック/レジスタファイル32、パレルシフタ30、ALU28及びフラグ及びシフトレジスタセット44及び46の側面に設けられる。更に左側ポート170はデータバスの入力及び出力の末端に設けられている。好ましい実施例ではないが、右側ポートを用意する選択も可能である。このあるプロセッサの右ポートは、隣接するプロセッサの左ポートに接続する為に使用される。例えば、プロセッサPR3の右側ポートは、プロセッサPR2の左側ポートに接続される。リテラルレジスタ174もまた定数を発生させる為に提供される。入力はメモリマツパー34から接続する左ポート170に接続され、右側ポート172からの出力もメモリマツパー34を通つて接続される。

アドレス線及びデータ線156及び158を多重構造にし領域を節約することができる。ラウンドロビン仲裁線160は4つのプロセッサPR0-PR3、4つのメモリモジュール60、及び外部データポート72及び73の間を花輪のような形で連結しているので、データバス56へのアクセスは有効に仲裁が行われる。4本のソース線162は、現在転送中のメモリアクセス要求のソースであるプロセッサ又はモジュール又はデータポートを特定している。その代わりに制御線164の状況に従つて、ソース線163を使つて、どのメモリモジュール60がメモリモジュールによつて次にサービスを受けるのかを示している。10本の割当て線168は、どのプロセッサ、モジュール又はポートがデータを受けるかを示す。ソース状況線166は(もし低論理であれば)ソース線162は更にプロセッサ、モジュール又はデータポートのどのメモリの要求が次にサービスを受けるのかを示す。ソース状況線166が高電位状態にあるということは、ソース線が現在のソースを

示していることを要わしている。また、メモリの読み出しに使用されるか又はメモリの書き込みに使用される為バスが停止状況 (Idle) にあることを示す2本のバス状況線165も提供されている。

第17図の流れ図はさらにデータバス56内の線の利用をさらに明らかにしている。一度、いずれかのソース (プロセッサPR0-PR3、RAMメモリモジュール60、又はデータポート72又は73) がデータバスへのアクセスを受けとると (即ち、一度、そのソースが花輪状連結線160から高電位信号を受けとると) ソースは線162上に4ビットロードを示す。ソースがRAMメモリモジュール60であつて、対応するメモリスケデューラユニット68が未解決のメモリ要求を有している場合、ソース線162はどのプロセッサ又はメモリモジュールが次にサービスを受けるかを示すようになり、線166は電位が引き上げられる。この後で、また未解決のメモリ要求を持つメモリモジュールがソースでない場合でも、メモリアクセス割当て線168の電位が引き上げられて

読出されるべきデータに関する10個の割当てから成るいずれかの組合せを示す。連動回路 (Interlock) は、データの完全性を保つ為に提供されている。もし割当て先に未解決のメモリ要求を持つメモリモジュール60を1つまたは2つ以上が含まれる場合、ワイヤーANDED線であるビジー線164の電位が引き上げられ、同一のメモリスペースに対し同時に読みこみ及び書き込みが行われることがないようにしている。

第19図は、RIC内のRAMシステムの組織を総体的に示す概略図である。各々のプロセッサPR0-PR3は、メモリマッパ34を介して、データバス56へのインターフェースとして働くバス制御ユニットBCU58に接続される。各々のBCU58は対応するメモリスケデューラユニットMSU68にも接続され、BCU58及びMSU68の各々は両方とも対応するRAMメモリモジュール60に接続される。

プロセッサが自分のメモリモジュール60にアクセスする時、プロセッサは自分のBCU58を

通つて自分のMSU68に接続される。MSU68は未解決のメモリ要求があるか否かを決定する。未解決のメモリ要求が存在しない場合、アクセスがただちに発生する。アクセスが決定されていない場合、MSU68はメモリのサービスを最初に要求したかを示すタグ (tag) を待ち行列にする。MSU68は先に来たものから先にサービスを行うスケジュール方針に従つて要求を待ち行列にする。特定の要求が列の先頭まで達した時MSU68がこのことをプロセッサ又はデータポートに信号で知らせる。このプロセッサ又はデータポートは再び要求を発生し、メモリアccessがただちにされる。プロセッサがそのプロセッサ以外のメモリモジュールにアクセスする時、バス制御ユニット58はデータバス56を介して通信が可能になるよう構成されなくてはならない。故に例えばプロセッサPR3がプロセッサPR1に相当するメモリモジュール60にアクセスする場合、PR3に対応するバス制御ユニット58は、プロセッサPR3をデータバス56に接続するよ

うに構成され、PR2に対応するBCU58は、信号をデータバス56によつて転送できるように構成され、プロセッサPR1に対応するBCU58は (プロセッサPR3に向かう方向で) データバス56をプロセッサPR1に対応するRAMメモリモジュール60に接続するよう構成されなくてはならない。上記のように、プロセッサはマザラウンドロビンのオーダーでスケジューラされた共有のデータバス56にアクセスできるまで待機している。プロセッサはバスにアクセスした後で、プロセッサはメモリ情報及び割当て先メモリモジュールを示す割当てタグを連送する。各々のプロセッサ又はデータポートは、一度に1つだけ未解決のメモリ要求を持つことができるので、どのメモリモジュールも最大で6つまで未解決の要求を持つことができる。各々のプロセッサのメモリインターフェースは、データバス56を通つて送られてくるメモリ制御信号を監視する回路を有している。この機能はメモリマッパ34によつて実行される。

RICの内部RAMメモリは、好ましくは最小の微細加工特徴が1ミクロン(ラムダ=0.5ミクロン)である(バストライバーのようなCMOS技術を実現する高出力構成素子を用いた)NMOSRICの中に作られた16KバイトのダイナミックRAMである。

故に、BCU58は全てのプロセッサをそれぞれ隣接するメモリモジュール60にアクセス可能とする一方、各々のプロセッサをデータバス56によつてこれ以外の離れたメモリモジュールにもアクセス可能にするという重大な役割を果たしている。BCUの構造は第20図及び第21図に示す。各々のBCUは、3つの両方向に導通可能な(バイダイレクショナル)スイッチ178、180及び182を有している。短いバス184はBCU58とそれぞれの対応するプロセッサのメモリマップ34を接続している。第2の短いバス186は、対応するモジュール60のアドレスレジスタ62及びデータレジスタ64に接続される。バイダイレクショナルスイッチ178はこれらの短い

バスの両方に接続され、バイダイレクショナルスイッチはある位置で単にメモリマップ34をレジスタ62及び64に直接接続するサービスを行つていて各々のプロセッサはそれに対応するメモリモジュール60に並列にアクセスすることができる。他の位置のバイダイレクショナルスイッチ178は、MSU68を(短いバス186を介し)データバス56に接続するサービスを行つている。同様に、スイッチ182は、(対応するプロセッサがその対応するメモリモジュールにアクセスしている時)データの流れをブロックするか又は、単に短いバス184をデータバス156に直接接続するか又は、短いバス184をバイダイレクショナルスイッチ180が封鎖モードである時に絶縁されている2つのデータバス56の組に方向を指定して接続するか、これらのいずれかを行つている。このオペレーションモードは第21図に示されていて、バイラインモードでのRICのオペレーションが可能にしてある。この場合、各々のプロセッサはオペレーションの前段階から

データを受けとり、同時にオペレーションの次に続く段階にデータ出力の流れを与える。故に、バイラインモードは、隣接するプロセッサの個別な組の間を異るデータの流れを並列に転送させてゆくことを必要とし、この機能は、バイダイレクショナルスイッチ180をブロックモードにすることによつてまたバイダイレクショナルスイッチ182を方向を指定して接続することによつて実現される。

プロセッサが離れたメモリモジュールにアクセスする時、異なる3つの構造から成るBCUが必要とされる。例えば、プロセッサPR3がプロセッサPROに対応するメモリモジュール60にアクセスする場合、PR3に於るBCU58はPR3をデータバス56に接続しなければならないし、PROに対応するBCU58はデータバス56に対応するレジスタ62及び64に接続しなくてはならず中間のBCU58は単にデータバス56のクリアオペレーションを可能にする必要がある。本実施例に於て、BCU58に必要な調整はデー

タバス56内のBCUモード制御線167によつて行われる。いずれかのプロセッサがデータバスを制御する時、そのプロセッサは更にBCUモード線167を使つて全てのBCUを制御することができる。仲裁線160がプロセッサがデータバス56を制御していることを示す時、全てのBCU58に於て満たさなくてはならない、唯一の必要条件は、バイダイレクショナルスイッチ180がデータバス56をブロックしていないということである。更に、各々のBCU58内のバイダイレクショナルスイッチ178は、データバス56をレジスタ62及び64に接続するように作動している。故に各々のアドレスレジスタ64は、要求されたアドレスを受取り更に適当なMSU68がその中のメモリモジュール60内の制御レジスタ70を作動させて必要なデータを供給させている。

BCU58を制御する上記のシステムはできうる範囲で最も有効な例というわけではないが、追加の制御線の必要性を最小にしてプロセッサ間におこる干渉を防止している。好ましくは、各々の

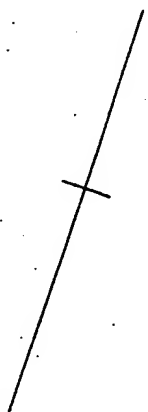
MSU68は直接データバスのバス状況線165に接続され、RAM制御レジスタ70が必要とする読出し／書込み及び作動ビットを受取っている。各々のMSU68は対応するプロセッサに対し直接接続する2本の線を持つて構成され、RAM制御データを与えてデータバス56を使用せずに地域的なアクセスを行っている。

メモリスケデューリングユニット(MSU)68では、第22図で示すようなオペレーションが行われる。データバス56がアクティブであつて、BCUモード制御線167がアクティブにされた時はいつでも、MSUはアドレス線156の先頭ビットを見てアクセスがこのMSUに制御されるメモリモジュール60に対するものであるか否かを決定している。もしそのメモリモジュール60に対するものであれば、次は、メモリの要求が未解決となつているか判断する。この場合、最後の要求を出したプロセッサまたはデータポートは、先に入つたものが先に処理される形式に配列される。最大で6個のソース(4つのプロセッサ

及び2つのデータポート)を持ち行列に並べられる必要があるので、小さなシフトレジスタがこれを実行する。同様にしてもし他のメモリの要求が既に未解決のままである場合、地域的なプロセッサによつて要求されるメモリアクセスは延期され、ビジー信号がもどつてくる。メモリ読出しが実行される時は、データバスが使用可能になるのを待つてから、データワードを割合て線168を通つて指示された割合て先に転送しなくてはならない。書込みが実行される場合には、帰帰信号は必要ない。どちらの場合でも、MSU68はただ待ち行列レジスタから次の未解決のメモリの要求を引きだし、そのソースに信号を送つてメモリアクセスが現在可能であることを知らせる。MSU68は次のソースが要求を再び発生するまで待機し、それから上記のように適当なメモリアクセスを行わせる。

当分野に通常の知識を持つ者には明らかなように、アドレスレジスタ64、データレジスタ62及び制御レジスタ70はMSU68とそれに対応

するメモリモジュール60の間のインターフェースとして使用される。



本実施例において、各々のメモリモジュールは、16ビットアドレスでアドレス指定される。これによつて、4つのプロセッサの各々に対し直接アドレス可能な領域を場合によつて64Kバイトまで増やすことが可能である。しかしながら、プロセッサは2種類の型のアドレス、即ち16及び32ビットを維持している。プロセッサ自身のメモリモジュールに直接アクセスする為には16ビットのアドレスが使用される。32ビットアドレスは外部メモリへのアクセスの為使用される。他のメモリモジュールにアクセスする場合は、プロセッサは16ビットのアドレスを送り、所望の内部メモリモジュールと結合する割合て信号を与えることでメモリモジュールを指定する。32ビットのアドレスは、プロセッサの制御に従つて、外部アドレス又はマッピングアドレスのうちいずれかとなる。外部アドレスとなるようアドレスが指示される場合、アドレスは外部メモリインターフェースに送られてプロセッサが実行される。さもなければアドレスはメモリマッパーに送られる。メモ



リマッパはアドレスが内部のものか外部のものを決定する為連合して探索を行う。内部のものであれば、関連した内部アドレスは外部メモリアンターフェースに送られる。

(状況バスのエンド・アラウンド・ループ54を含む)状況バス52は4つのプロセッサPR0-PR3及び外部状況ポート74及び75を相互接続している。状況バス52は桁上げ状況線190、桁上げ入力出力線191、オーバーフロー線192、負数線194、ゼロ線196、プロセッサ同期線198及びシフト循環線199から成る7本の線のみを有している。故に、エンド・アラウンド・ループ54を含めても状況バスは、たつた14本の広さしかない。各々のプロセッサは、プログラム可能なスイッチである対応する一対の状況マルチプレクサ48又は50を有している。これらのスイッチは、選択的に且つプログラムに沿って各々のALU28の状況出力又は入力線を状況バス52及びエンド・アラウンド・ループ54の両方を含む状況バス線に、いずれかの方向を指定して接続

している。第23から第25図は、状況マルチプレクサ48及び50で選択された状況バス接続のいくつかの例を示している。プロセッサが独立して動作する時、第23図で示すように状況線の接続は必要とされず、状況マルチプレクサ48及び50は状況バス52及び54に開路を作りだす。隣接するプロセッサがパイプライン構造で動作する時、(例えばPR3の)データ出力がPR2の人力として与えられている場合、プロセッサ同期線198のみが接続される必要がある。最後に、プロセッサPR3及びPR2が32ビット又はそれ以上の規模のプロセッサの一部としてロックステップ構成で動作される場合、プロセッサPR2の状況出力は、全てプロセッサPR3に対する状況入力として接続される。プロセッサ同期線は、ロックステップモードで使用され現在のオペレーションが完了する前には確実に新しいオペレーションが開始されないようにしている。例えばロックステップ構成のプロセッサがアクセスしている全てのメモリモジュールに対し、他のソースから

の未解決なメモリ要求が不均一に配分されることによつて、全てのプロセッサは、そのメモリアクセスを同時に受けとることはない。状況マルチプレクサ48及び50によるプログラム可能な相互接続の状況は、DLA22から接続される制御線によつて管理される。状況線190から196は、ALU28及びフラグレジスタ46からの状況入力及び出力に接続され、シフト/循環線199はバレルシフタ30のリンクイン及びリンクアウトビットに接続され、プロセッサ同期線198はDLA22に接続される。外部状況ポート74及び75も同様のプログラム可能な接続回路を有している。

本発明である再構成可能な多重プロセッサ機能を提供する為共動する上記で示したような構造の特徴はこれからさらに詳しく説明する。

第26、27及び28図は、本発明である再構成可能な集積回路のオペレーションの3つの主要なモードを示している。第26図は、全体的に独立するモードの組織を示している。このオペレー

ションではプロセッサPR0-PR3は単一チップ上に配置された実質的に4つの独立するプロセッサとして働いている。4つ別々の命令の流れが別個のプロセッサPR0-PR3に与えられ、独立するプロセッサ間に必要とされる唯一のインターフェースは、制御バス及びデータバスプロトコルによつてアクセスされるものである。

同じ構造はまた配列処理オペレーションを行えるようになつている。配列処理に於て、プロセッサ間の相互接続は独立モードの場合と同じである。配列処理と異なる点は、各々のプロセッサが、同じ命令の流れを受けとつているということである。もちろん配列処理はまたプロセッサのロックステップで接続されるプロセッサの2次的な組(例えば2つの32ビットプロセッサ)に基づき構成される。

第27図は内部的なロックステップモードで動作されるRICの例を示す。この例では、4つ全部のプロセッサが再構成されて単一の64ビットプロセッサとして動作するようになつている。こ

の構成では、制御バス14が単一の共通な命令の流れを運んでいて、この命令の流れを全てのプロセッサが受け取る。上記のように必要な同期桁上げ及び状況ビットは状況バス52及び54によつて通信が行われる。更にシフト及び循環連結が設けられているので望みのビット又は循環オペレーションをどれでも64ビットワードでデータバス56を介し実行することができる。(この連結に関しては、データバス56は、上記で示したようにパイプラインモードで構成される。)1ビットシフト及び循環がシフト/循環線199によつて実行される。RICをこのモードに正確にプログラムすることによつて、各々のプロセッサが64ビットワードの16ビットセグメントの読出し又は書込みを行い全てのプロセッサが4つのRAMメモリモジュールの対応する位置でこの読出し又は書込みを同時に行うので、単一クロックサイクル内においてRAMメモリ66に対する64ビットでのアクセスが可能となる。

第28図は、パイプラインモードで作動するよ

う構成される。更に、ロックステップ及びパイプライン構造を組合せることも可能であるので、RICは例えば32ビットプロセッサを接続して2段パイプラインを作つて作動するようにもできる。この場合、次々とパイプラインで接続されるステージの間でのデータ転送はデータバス56で多重にして行われる必要がある。故に、この実施例では、プロセッサPR3及びPR2が2段の32ビットワードパイプラインの1段目を形成する場合、プロセッサPR3はプロセッサPR1に対応する割合で線168の電位を引き上げながら第1段の出力の最上位ビットを送信し、続いてPR2がプロセッサPR0に対応する割合で線168の電位を引き下げながらデータ線158を使つて出力ワードの最下位16ビットを出力する。

再構成を行う為には、3つの主要な型の割り込みが使用される。第1の型の割り込みは、他のプロセッサ(即ち、割り込み先(Interruptee))の資源にロックステップモードのオペレーションを設定するように要求する。このモードでは、1つのプロ

う構成されたRICの例を示す。この構成において、4つのプロセッサは、もともとは単一のデータの流れてあつた連続するオペレーションを実行している。即ち、全体的な命令の組は、(プログラマーによつて)4つのオペレーションの組に分割される。この4つのオペレーションの組はそれぞれほぼ同じ時間を必要とする。そこで4つのプロセッサは各部分が連続するデータの流れのうち、分割された命令プロセスの一部分のみを各々で実行する。故に、このようにして小分けされた命令の組を用いてデータの流れて操作しなければならない場合、スループットは4倍になる。命令の組はプログラマーによつて4つの命令のサブセットに分割されるので、チップレベルでは、命令の流れは単に1つ1つがパイプラインモード内の各々のプロセス段階に向けられた4つの別々で個別の命令の流れとして現われる。上記で示した通り、BCU58はデータバス56を小分けして、パイプラインモードで連続するステージの間(即ち、隣接するプロセッサの間)をデータが直接の転送

セッサが共働単位の中の他のプロセッサのオペレーションを支配していて、故にこのプロセッサはロックステップでの共働単位内のマスタープロセッサと呼ばれる。この型の割り込みのプロセスを(プロセッサがコンテキストスイッチを実行した後)すぐに実行することを全てのプロセッサが受理する場合以外は、割り込みはこれ以上の動作をおこすことはなく、マスタープロセッサによつて後で改めて発生される。この場合、自分の割り込みが拒絶されたマスタープロセッサは、そのプロセスをバッファの中に入力しておいて、さらにスケジュールを変更し、次のスケジュールによるプロセスの実行を開始する。その後再度割り込みを送ることもマスタープロセッサの役割である。故にロックステップでの割り込みは、すぐにプロセスが実行可能な時のみ受け入れられる。「どこそこのプロセッサが必要とされている」「自分のプロセッサは、今は使用可能である」「自分のプロセッサを解除せよ」及び「どこそこのプロセッサが解除される」と等といったオペレーションを意味するプロセッサ

間での割込みの必要性を最小限にして、このプロトコールはプロセッサ間の通信を最小にしている。これによつて割込みに使用されるバスサイクルの数が減つてプロセス及び割込みのスケジューリングが複雑でなくなっている。単に拒絶された割込みをバッファしたり、延期したりすることは、上記で示した選択しうる例に比べて複雑なプロセスではない。

ロックステッププロセスでの割込みの為に要するコンテキストスイッチは他の型の割込みコンテキストスイッチより少なくてすむ。プロセッサがロックステップに構成される場合、マイクロシーケンサ36は、ロックステップの全期間中は停止状態(idle)に保たれる。実際のところ、マスタープロセッサのマイクロシーケンサ36が他のプロセッサ内のマイクロシーケンサの機能を代行する。故に、この場合シーケンサは働かなくなるので、マイクロシーケンサのコンテキストスイッチはそのままでよい。もちろんこのことは真のマイクロ割込みが許可される場合のみ問題となる。即

当なコンテキストスイッチを実行する。マスタープロセッサは、割込みのソースに通常の割込み応答を用いてそこから発生された割込みがただちに処理されることを知らせる。マスタープロセッサは、また、割込みを必要としない共働単位内の全てのプロセッサを解除する。マスタープロセッサは更に再びスケジュールを行う為に割込まれたプロセスをバッファに入力する。マスタープロセッサは共働単位内の他のプロセッサの割込まれたプロセスを再び続行させる。共働単位内の全てのプロセッサが再び作動可能になつた後で割込まれたロックステップのオペレーションが再び続行される。

ロックステップの共働単位内で作動している従属プロセッサがロックステッププロセスより低い優先順位を持つ割りこみを受けとる場合、従属プロセッサは、低い優先性のプロセスをバッファし、後にスケジュールし直す。故に、マスタープロセッサが、共働単位内の他のプロセッサのいずれかに向けた割込みを確認する必要がある間は、ロックステップより低い優先順位を持つてい

ち従来装置におけるマクロ命令に相当するものを実行する間に割込みが発生可能である。本実施例では、マイクロ命令を省略すれば複雑さを減少できるが、これを使うことも可能である。マイクロ命令を使用する意義には、2つの要素が存在する。第1は、大規模CPUに多重チップを使用する場合のような応用例では、その性能を最大限に利用する為にマクロ命令が使用されないことである。第2は、あるマイクロ命令はストリング操作のように非常に長くなるということである。

ロックステップ構造の中の従属プロセッサ(slave processor)の割込み管理回路は自分を指示する割込みの存在を知る為に制御バス14の監視を続けている。しかしながら、もし、ロックステップで構成される共働単位内のいずれかのプロセッサがその共働単位内で現在実行しているプロセスより高い優先性を持つ割込みを受取る場合、従属プロセッサに向けられる割込みを監視しているマスタープロセッサは、ロックステップで構成される共働単位内の全てのプロセッサにおける適

る従属プロセッサの割込みに対し、マスタープロセッサの割込み管理回路はいかなる機能も実行する必要がない。それぞれの従属プロセッサ内の割込み管理回路12は低い優先順位の割込みをバッファする。

第2の型の割込みは割込み先プロセッサが実行する、いくつかの計算を特定することを要求する。(即ち、これはよく知られる、古典的、な割込みである。)この型の割込みの次には、実行すべきプロセスを特定する命令の流れが与えられる。この型の割込みは、その優先順位に従つてプロセスされる。このような割込みがサービスの為にスケジュールされた時には、割込みの後で与えられた第1番目の命令に従つてコンテキストスイッチを行っている。

第3の型の割込みは、パイプラインプロセスを開始させる。パイプラインでの割込みの管理は、ロックステップ型での割込み管理と同様である。マスタープロセッサがパイプラインの割込みを送り出すと、これを受けとつた全てのプロセッサは、

自分がパイプラインプロセスに参加できるかを信号で知らせる。受取つたプロセッサが全てそろつて参加できない場合、割込みは無効となつて後で再び続行される。同様にパイプラインプロセスにおけるマスタープロセッサは、共働単位内のいずれかのプロセッサに向けられる割込みを監視してパイプラインプロセスより高い優先順位を持つ割込みを実行している。従属プロセッサは低い優先性の割込みのみをバッファする。パイプラインでの割込みの制御とロックステップでの割込みの制御との間の主な違いは、パイプライン割込みプロセスでの従属プロセッサがそのそれぞれの持つマイクロシーケンサ36を使用し独自のマイクロ命令の流れを実行していることである。故にそれぞれの従属プロセッサのマイクロシーケンサ36内に含まれるコンテキストは保持されたままである必要がある。

資源が共用されている場合、前に述べたようにアドロックの発生は、非常に危険であつて必ず避けなければならない。例えばプロセッサPR1

及びPR0がそれぞれプロセッサPR2及びPR3、両方の資源を使いたい場合、またプロセッサPR0がプロセッサPR3を支配しようとしている一方でプロセッサPR1ができ続きプロセッサPR2を支配している場合、もしもプロセッサPR1及びPR0がそれぞれの可動できないプロセッサPR2又はPR3が可動できるようになるのをただ待つようにだけプログラムされているとすると、デッドロックが起こりうる。本発明においては、いくつかの手段を講じてデッドロックを防いでいる。第1に各々の実行中のプロセス及び各々の割込みには、プログラマによつて独得な優先順位が与えられる。概念的には、各々のプロセスにそのプロセスの優先順位及びプロセスを開始させる割込みの優先順位と同一の名前をつけることができる。システム内で実行中の全てのプロセスは、異なる優先順位をそれぞれ持つ為、より高い優先順位を持つ割込みが割合てプロセッサを先取るので行き詰まりは起こり得ない。本実施例に於ては、1回の割込み期間中には、アドレスバス

の8本の低いオーダーの線に示される256の優先順位がシステム内に存在できる。割込みが認められバッファされる時はいつでも、優先順位に関する情報もバッファされる。バッファされた割込みがサービスの為にスケジュールされる時、8ビットの優先性に関する情報を含む割込み制御ワードがとりだされプロセスを開始させる。

故に、全てのプロセス(即ち、命令の流れ)がゼロから255の間の優先性レベル(又は名前)を有する。この場合255が最も高い優先順位になる。どのプロセスもプロセス自体と同じ優先順位を持つ割込みによつて開始される。多重プロセッサプロセスにおけるマスタープロセッサは、同じレベルの割込みによつて全ての必要とする追加のプロセッサに割込みを行う。ロックステッププロセスでは、同じマイクロ命令の流れが全てのプロセッサによつて実行される。ロックステッププロセスにおける各々のプロセッサは他の全てのプロセッサとまったく同じプロセスを実行しているものとして扱われる。(しかし、桁上げビットの

存否といったロックステップ内の相対位置によつて時々わずかな違いはある。)ロックステップで接続するプロセッサ間における違いは、これらプロセッサの位置がプロセッサ間のハードウェアの相互接続を決定していて、マスタープロセッサが共働単位内の全てのプロセッサに対しより高い優先順位の割込みを実行していることである。パイプラインプロセスでは、各々のプロセッサは、同じ割込みで割込みが行われ、もともとは各々のプロセッサは同じ命令の流れを処理しパイプラインプロセス内の各々のプロセッサは、異なるマイクロ命令の流れを実行することができる。次に、プロセッサは元の共通なマイクロ命令の流れをそれぞれの持つ場所の数によつて分岐し、それに対応する適当な分岐を選択し各々のプロセッサに対する別々のマイクロ命令の流れを作る。パイプラインモード内の各々のプロセッサは別々のマイクロ命令ルートをを行うことができても、全てのプロセッサは、チップを管理するという目的では同じプロセ

スを実行していると考えられる。パイプラインでの共働単位内のマスタープロセッサは、ロックステッププロセスと同様に単位全体に対しより高い優先性を持つ割り込みが処理される。1つのプロセッサのみを必要とするプロセスは、多重プロセッサでのプロセスを管理する為に必要としたプロトコールは必要としない。

上記で示した優先づけ方法によつて生ずる拘束としては、制限なしにはプロセスを共用することができないことである。この制限とは、同一プロセッサ内で共用されるプロセスは一度に2回以上の発動は不可能であることである。この制限は、いかなるプロセスにも正式にあてはまるが、多重化した別々のプロセスが共用のルーチンを開始することができるので、この制限は、共用のルーチンに関してのみ重要である。この拘束によつて一つのプロセスの為に複数の状況を記憶する困難性を回避することができ、同じプロセスが多重に発動されたことを見分ける為に必要となる機構の必要性をなくしている。ルーチンを共用して循環を

可能にする望ましいこの特性は、この後に述べる方法によつてほとんどすべて実現できる、ルーチンが共用されるかあるいは回帰的に呼出される場合このようなルーチンが発動されるごとに特有な優先性レベルが与えられる。この解決法は重複する共用ルーチンに必要な制御を記憶する領域を節約することができる。このようなタスクを開始するにあつては、優先性レベルが制御ワードへのアクセスを起こす。制御ワードにはルーチンの最初のアドレスを指示する区域がある。共用又は回帰的ルーチンは、全て制御記憶回路38内の同じ開始アドレスを指示する多重な優先性レベルを持つことになる。

256の優先性レベルを用意すればたいの応用例には充分である。故に本実施例ではこれを採用している。しかし、もつとたくさんの優先性レベルを用意することが望ましい場合、例えば8ビットによる優先順位の特定制より10ビットを使うことはもちろんとるにたらない問題である。

256の優先性レベルを用意することは1度に

256のみの有効なプロセスが存在することを意味する。有効でないプロセスは数に制限なく存在可能である。

例えば本発明による再構成可能なICを用いて32ビットCPUに対抗することを仮定する。再構成可能なICにおけるプロセスは以下のようなタスクが与えられている。即ち命令とり出し、コンテキストスイッチ、命令間の対抗、外部割り込みレベルごとに1つのプロセス、自己テスト、リセット等である。この場合256のプロセスで適しているように思える。

各々のプロセッサ内の割り込み管理回路12の主なタスクは、割り込み信号プロトコールに従つて信号を受取りまた発生することである。割り込み管理回路12は割り込み信号を送信及び受信していて、新しい割り込みが受けとつた時はいつもスケデューラ16に信号を送り、現在実行中のプロセスの優先順位と比較した新しい割り込みの優先順位を知らせる。スケデューラ16及び割り込み管理回路12は両方とも現在実行中のタスクの優先順位を記憶

している必要がある。新しい割り込みが現在実行中のプロセスを続行させる時、スケデューラ16はただちに新しいプロセスがシーケンサ36の内容を含めた全体的コンテキストを記憶することを必要とするか否か又は、部分的なコンテキストスイッチのみを必要とするかを確認しなくてはならない。必要なコンテキストスイッチについての情報は、高いオーダのアドレスビット84でコード化される。優先順位の特定制には8ビットしか必要としないので必要なコンテキストスイッチの程度といつたような割り込みに関する追加情報をコード化する為に6ビットが使用される。受取つた割り込みの優先順位がより低いものであつてその為パツファされる場合、このようなプロセスは必然的にロックステッププロセスでないで、パツファされた優先順位が最終的にスケデューラされる時には、全部のコンテキストスイッチが必要となる。ロックステップのプロセスだけは、全部のコンテキストスイッチを必要としない。このような場合、スケデューラは適当なコンテキストを記憶し、現

在のタスクをバッファし、プロセッサが新しいプロセスを開始する用意ができたことを割込み管理回路12に知らせる。

スケデューラ16の構造及びオペレーションは、第29図を参照してこれからさらに詳しく説明を行う。スケデューラ16は優先順位によつて有効なプロセスのリストを作っておかなくてはならない。有効プロセスとは、スケデューラすることのできる最も高い優先順位を持つ時、実行を開始するプロセスである。ハードウェアを節約する為に本実施例では、1ビット毎の256のリスト即ち、有効なタスクを記憶する為の有効プロセスバッファ200を保持している。特定のプロセッサが1と名づける有効プロセス(即ち1の優先性を持ち実行を待つプロセス)を有している時はいつでも、スケデューラ16の中のこのプロセッサに対応する有効プロセスバッファ200の1番目の位置に1含まれている。有効プロセスバッファ200内のプロセスは実行の為のスケデューラが行われる時には、バッファ200内に於るこの位置(これ

はこのプロセスの名前であり優先順位である。)を使つて各々の有効プロセスのコンテキストを示すポインタを持つテーブル220にアクセスしている。故に、バッファ220には2つの主な目的がある。それは有効プロセスを監視することと有効なジョブを優先順位でランクづけすることである。故に、スケデューラ16は5つの主要な構成部を持つ。即ちバッファ200現在実行中のプロセス優先順位の為のレジスタ202、次の有効プロセスの優先順位の為のレジスタ204、プログラム可能タイマー206及び線形的にバッファ200を調査し次に高い優先順位をテストし、見つけるリニア検索論理208である。

スケデューラ16のオペレーションは第30図の流れ図で示される。スケデューラ16は、割込みの発生、有効なプロセスのブロック、又は、プロセスの完成の3つの起こりうる原因(勝発要素)の1つによつて作動されるようになる。ブロックされるプロセスとは、必要な資源が使用できない為実行の続行を中止させられるものである。例

えば要求された資源を全部は受けとらなかつたロックステッププロセスは、他のプロセスから(受けとらなかつた)情報を必要とするプロセスであるとしてブロックされる。プロセスがブロックされると、プロセスの実行はまず停止する。次に全ての必要なコンテキストが記憶される。更に、スケデューラ16は、有効プロセスバッファ200内のブロックされたプロセスの優先性に匹敵する位置に1を入力することによつてこのブロックされたプロセスはバッファされる。ブロックされたプロセスの優先順位は、次の優先順位の為のレジスタ204(フローチャートにおける「次の優先性」の項)にロードされるのでブロックされたプロセスは少くとも次にスケデューラ16が呼び出されるまで停止される。

ブロックされたプロセスは、もし必要であればタイマー206を使つてプログラムすることが出来る。このようなプロセスがブロックされる時、プロセスは命令でタイマーをプログラムする。タイマー206がゼロまでカウントダウンする時、

スケデューラ16を作動させる勝発要素が生じる。この要素によつてスケデューラ16は通常のスケデューラサイクルを実行するようになる。タイマー206をセットしたブロックされたプロセスが最も高い優先順位を持つていれば、スケデューリングが行われる。しかしながら制御パス14から割込みを受けとる場合、又は他のプロセスがブロックされる場合又はプロセスが完了した場合にはタイマー206は作動しなくなる。このような場合には通常のスケデューラサイクルが開始され、故にタイマー機能が不要になる為タイマー206は作動しなくなる。このブロックされたプロセスに対するスケデューリング法は、一方ではブロックされたプロセスに引き続き、チップの資源を使用させいつ必要な資源の用意ができるか判断させていて、また他方ではより高い優先順位を持つプロセスを必要な時間より長くブロックさせている間を調整するものである。この調整は低価格のハードウェアですみ、良好なスケデューリング動作を得ることが期待される。しかし、現在のプロセ

スの実行を脱行しつつ、現在のプロセスより高い優先順位を次のプロセスが持つようにすることは、スケデューラ16のオペレーションをいくらか複雑にすることになる。

スケデューラ16が割込みによつて呼び出される時、(レジスタ202から入力された)現在のプロセスの優先順位及び(レジスタ204から入力された)次のプロセスの優先順位の高さと共に受取つた割込みの優先順位がコンパレータ212によつて比較される。新しく受取つた割込みが3つの中で一番高い優先順位を持つ場合のみ先取が起こる。先取する必要がある場合、現在のプロセスを停止させコンタクトスイッチが実行された後でだけ新しく受けとつた割込みが開始される。スケデューラは次に次のプロセス及び現在のプロセスの優先順位の高さを次の優先順位の為のレジスタ204に入力する。有効プロセスバッファ200内の対応する優先性レベルに1を書きこむことによつて現在のプロセスがバッファされる。先取する必要がある場合、新しく受けとつた割込

みは、有効プロセスバッファ200にバッファされ、バッファされた割込みに関する情報はこれ以後受けとり側のプロセッサに記憶される必要はない。次の優先順位と割込みの優先順位のうち高い方が次の優先順位の為のレジスタ204に入力される。故にある場合では次の優先順位の為のレジスタ204が有効な情報を内容として持たないようにすることができる。これは、有効プロセスバッファ200によつてリニアな検索を行い次に最も高い優先順位を持つプロセスを発見することが時として必要になるからである。必要とされる時に次の優先順位の為のレジスタ204の内容が有効でなかつた場合、スケデューラ16のオペレーションは他に有効なプロセスが発見できない場合も含めて有効なデータが見つかるまで停止する。最終的にプロセスを完了することによる作動をおこす原因が受けとられると、(これはプロセス状況ビットを通してスケデューラ16に知らされる)次のプロセスが開始し、次に高い優先順位を持つプロセスの検索が開始される。

割込みシステムのオペレーションを明らかにする為、割込み制御ワード(ICW)のフォーマットを以下に説明する。各々のプロセスは、それと共働するICWを有している。ICWはプロセスの優先順位に予め規定される定数をつけることによつてアクセスされる。全てのICWは、制御記憶回路38内の256ワードテーブルの中に含まれる。あるICWをアクセスする為につけられる定数とは本質的には、256のワードページに対するページフレーム数である。ICWは、第31図で示す3つのフォーマットに規定される。もちろん必要に応じて他のフォーマットを規定することもできる。1つのICWにはプロセスを開始又は改めて開始させる必要とされる割込みの優先レベルに相当する必要な情報が含まれている。

ICWのタイプフィールドは命令オPCODEに類似する。タイプフィールドは4ビットであるが、3つの型のICWが規定されている。タイプ0は単一プロセッサによるプロセスの為であり、タイプ1はロックステッププロセスの為、またタイプ

2はパイプラインプロセスの為のものである。タイプフィールドは割込み管理回路に対する命令としても使用され以後の割込みの発生を命じている。

位置フィールドは、予定された割込みを受けとつたかを確認している。このフィールドには各々のプロセッサに対し1ビットが含まれていて各々のビット位置における内容を使用して、それに対応するプロセッサが割込みを受けとるべきであることを示している。

優先性マスクフィールドはプロセスの名前又は優先順位とは異なる実行の優先順位を特定する為に使用される。言い換えれば、優先性マスクフィールドが使用される場合、プロセスの優先順位は、(実行の優先順位というよりむしろ)せり合ひ時の優先順位(bidding priority)と考えることができる。PMビットは、優先性マスクが使用されるか否かを特定する為に使用される。優先性マスクは、多重に資源を利用する必要があるプロセスを完全に実行できる可能性を高める為に使用される。優先性マスクは、他のどのプロセスにも与

えられていない優先値を使うように限定されている。さもなければ、2つのプロセスが同じ優先順位で資源をとりあう時、アッドロックが起こりうるからである。

次の優先性フィールドは同一プロセスに多重優先レベルを与えることができる。このフィールドはNTビットが1である場合のみ有効となる。この機能によつて、ランタイムで規定される優先順位をプロセスに与えられる。この機能は、プロセスの非常事態がやがて変化する時に有効となる。例えばプロセスは、特定の優先性レベルを持つ割り込みを、必要なサービスを予期して発生することができる。割り込みが直ちに動作を開始させない場合でも、割り込みのイニシエータが後にプロセスを開始させる。しかしながら、時間的な遅れによつて(本実施例に於て)割り込みのサービスを受ける必要性がさらに急を要するようになるのでイニシエータは次に最も高い優先順位を持つ割り込みが発生可能になる。次に高い優先順位についての情報は次に高い優先順位の為のフィールド内に含まれ

ている。この機能を利用する別の方法としては同一プロセスの多重処理を開始する方法がある。このモードのオペレーションでは、ある優先性レベルから優先性レベルへと位置フィールドが変わる場合に、並列処理が可能となる。同じ位置フィールドが使用される場合には回帰的ルーチンを使用することができる。

最後に、制御記憶アドレスフィールドは現在の割り込みによつて呼び出されたプロセスマイクロローチンのマイクロ命令のアドレスを指示している。

前に述べたように、割り込みが送られた時には、14本の制御記憶アドレス線84のうちの8本の線のみが使用されて優先性がコード化される。故に、残る利用可能な6ビットのうち1ビットをコンテキストスイッチを全部実行するか又は一部実行するかの決定のために使用することができる。更にこれ以外のもう1ビットは割り込みがただちにプロセスを実行しない場合、割り込みをパツファするかどうかの判断のために使用される。故に第32図は割り込みが送られる時の制御記憶アドレス線84

のフォーマットを示す。

割り込みシステムのオペレーションをさらに明らかにする為に、第33図は、プロセッサPR3がプロセッサPR1に対し割り込みを発生した場合のサンプル割り込みに相当するタイミング表を示している。タイムnにおいてプロセッサPR3が制御バス14を支配していて、(線13上に)自分が割り込みのソースであるという信号を与えていて(線DR1上には)割り込み先がプロセッサPR1であるという信号を与えている。同じこのタイムに於ては現在の割り込みが確認されるまでは他の割り込みが制御線14を通つて転送されることはないのでプロセッサPR3は、割り込み準備線104の電位を引き下げる。次のクロック間隔では、プロセッサPR1は、割り込みバス11の適当な線の電位を上げることで割り込みの確認を行い、コンテキストスイッチを実行した後でプロセッサPR1が割り込みのプロセスを開始させたことを知らせている。次にプロセッサPR3は、割り込みによつて知らされたプロセスを開始させる為のマイクロ命令のア

ドレスを転送するアドレスバス84を操作する。プロセッサPR1がそのコンテキストスイッチの実行を完了すると、プロセッサPR1がこのマイクロ命令のアドレスをロードし、割り込みバス13及び割り込み準備線104の電位を引き上げて(又は引き下げを止めて)割り込みが再び受けとれる状態となつたことを知らせる。

ここまでは再構成可能な機能及び多重プロセッサ機能を提供する割り込みプロトコールの組織を説明してきた。再構成可能機能を提供する為に必要な命令構造に使用するこの他の構成要素は、DLA 22及びDLAデコード26によつて通訳される時に、状況バスマルチプレクサ48及び50の再構成を特定するか又は、BCU58の内の一つに含まれるバイディレクショナルスイッチ178、180及び182のオペレーションを特定する命令を有している。

本発明の再構成可能なICの実施例は、84ピンパッケージの中に含まれる。(64ピンでデータ及びアドレスをまかなう112ピンパッケージ



も選択できるがあまり経済的でない。) )

ある種の構成に関しては、84ピンパッケージをフルに必要としない。故に、このような応用例の為に設計されたR I Cを中間的な量で製造する為には、もつと安価なパッケージを使用することができる。

本発明のR I Cの為に好ましいパッケージは、例えばここに参照として示すエレクトロニクスマガジン1981年6月30日号の39から40頁で記載されるような84ピンプラスチック・リード・チップキャリアである。しかし第34図ではR I Cが超大規模D I Pパッケージの中に含まれる場合のピン配置を便宜上示す。

本実施例のI R Cは、2つの16ビットデータ/アドレスポートを有している。各々のポートはデータ及びアドレスを運搬する為の16本のバイダイレクショナル線を有している。ポート1におけるデータ/アドレスピンは、1から16まで、ポート2のピンは26から41までの番号がふつてある。データポート2の対応するデータポート1

ることができる。外部データポートには主として2つの機能がある。第1の機能は、外部データポート72及び73としての役目である。データポートの主要機能の一つは、内部データバス58と外部データポートピンの間のインターフェースとしての役目である。實際上これの意味することは、データポートが内部命令を受けとつてデータ及びアドレスを外部に送る機能を持つということであつて、データポートはこの機能を果たし、それを適当な信号にして外部ピンに送ることができる。第2にデータポートは、外部ピンから情報を受けとり、次に適当に変換して、適当にその信号を内部RAMバス58に送っている。第2の主要機能は、ブロック転送モードである。これは、例えば1つのアドレス及びデータワードを外部に送る命令を送るかわりに、内部的に制御及び命令を受け取る多重アドレス及びデータワードを送り出している比較的一般的な命令を送っている点で上記に説明した他のオペレーションとはわずかに異なっている。

のピンは、機能的に等しいのでポート1で動くピンのみに関し示す。第1及び第2図で示す外部データポートF2及びF3にデータポート1及び2が対応する。簡単なピンの配線に加えて、外部データポート72及び73は、1個以上のR I C、外部メモリ、I/O装置及びアドレス装置の間を結ぶ共有外部バスの制御に対し仲裁を行う和議信号(handshake signals)(ピン17及び18)を有している。各々のポートは、バスでのデータ及びアドレスの送信及び受信の同期を行う為の追加のピン(22及び23)を有している。各々のポートは、バスの状況、及びバイダイレクショナルな一対のプロセッサタグアイデンティファイヤー(24及び25)に信号を送るバイダイレクショナルな3本の線の組(ピン19-21)も有している。データポート1と2は独立している。しかしながら内部的には、両方のポートに同じオペレーションを回路的に実行させ、外部的には2つのポートを1つのポートのように扱うことでこれらのポートを単一の大きなポートとして作動させ

従来において、データ又はアドレスの内の最上位の固まり(chunk)は第1のサイクルで送られ、それに連続するサイクルにおいて次々とより下位の固まりが送り出される。全てのアドレス又はデータの固まりは最初の(最上位の)固まりが送られた同じポートから送られる。

これらの2つのデータポートはチップ上の全てのプロセッサに共通する。2つ以上のプロセッサを持ついかなるプロセッサ又はいかなる内部構成も、いずれかのデータポートを使用可能である。ポートは、プロセッサ内のメモリマップ34によつて選択される。

以下で述べる外部通信プロトコールに関する機能に加えて、外部データポート72及び73もプロセッサによるデータバスプロトコールの使用に関し、上記で説明したものと全く同じ回路を有している。外部プロセッサもまたチップ上のメモリにアクセス可能である。

データバスの仲裁 共有のデータバスを仲裁する機能が備えられている。各々のR I Cは、外部仲裁回路、マスタースレーブ機構、又はラウンドロビン仲裁機構で動作する。R I Cの動作が開始すると、仲裁モードが指示される。仲裁信号はデータバス可動(D P A V)及びデータバス容認(D P G R)信号と呼ばれる。信号はポート1に対してはD P A V1及びD P G R1(ピン17と18)、ポート2に対してはD P A V2及びD P G R2(ピン42と43)と表示がつけられる。以後これらの信号の表示は2つの同一なポートの間を区別する為に参照番号を使わずに参照する。外部仲裁回路モードが使用される時R I Cがプロセスを開始し、外部回路に接続するD P A V出力信号の電位を上げることによってバスを支配する。

仲裁機構がプロセッサの要求に対しデータバスが使用可能であるか判断し、要求元のD P G R信号の電位が引き上げられる。マスタースレーブモードでは、マスターR I Cが常にバスを制御している。従属R I Cがバスを使用したい場合には要

求をD P A V信号でマスターに送っている。マスターは、D P G R信号の電位を上げることでバスが使用可能であることを知らせる。この他のモードの仲裁機構としてはラウンドロビン機構がある。この機構では、論理1がデータバスを使用するプロセッサの間を循環している。R I Cがデータバスのオペレーションを完了しているか又は、続行中のものがない場合、R I CはそのD P A Rの電位を引き上げる。これは、隣接するR I CのD P G R信号に接続される。データバスオペレーションが続行中である場合、D P G R信号が1になると、R I Cがデータバスを支配するようになる。さもなければR I CはそのD P G R信号の電位を上げる。バスを使用できなかつたR I Cは他の資源がいずれもバスの使用を必要としていないと判断した場合にこのR I Cは循環している1をトラップする。そこで(もし存在すれば)同期を維持する為にこのR I CはそのD P G Rを引きあげて1を再び循環させる。

データバス状況: 各々のポートがデータバスの

状況を示す3つのピン(19-21及び44-46)を有する。データバスを制御している使用中のプロセッサはバイディレクショナルピンに出力してデータバスの状況を知らせる。状況ピンの配置は、ポート1に関しテーブル1にリストしてある。(ポート2はピン44-46が第34図のピン19-21に相当することを除けば同一である。)

テーブル1 データバスピン配置

ピ ン			機 能
19	20	21	
0	0	0	隣接するR I Cへの書き込み
0	0	1	隣接するR I Cからの読出し
0	1	0	割当てアドレスへの書き込み
0	1	1	割当てアドレスからの読出し
1	0	0	メモリへの書き込み
1	0	1	メモリからの読出し
1	1	0	I/O装置への書き込み
1	1	1	I/O装置からの読出し

データバスの状況はデータバスオペレーションの型式を示している。データバスオペレーションに関しては4つの割当て先が存在しうる。即ち、割当てられたR I C、データ転送の最初の部分によって特定されたアドレスによる割当て、システムメモリ又はシステムI/Oユニットの4つである。読出し又は書き込みオペレーションはこれら4つの割当てのいずれでも実行可能である。いずれかの割当て先に転送される16ビットアドレスユニット及び16ビットデータユニットの実際の数は、R I Cのプログラムによって又は、割当てのハードウェアによって決定される。直接的な通信は隣接する1つのR I Cに対してのみ有効である。割当てアドレスモードはデータバスオペレーションに対し1つ又は複数のソースを特定する一般的方法である。データ転送の第1の部分は1つ以上の割当てを決定するアドレスを特定している。割当てアドレスを含む16ビットユニットの数は、使用者によって特定される。このモードでは、アドレスは、データバス上の全ての受信プロセッサ

に送られる。受信側は、アドレスが自分の持つアドレスの1つであるかを判断する。全てのアドレスを受取ったプロセッサは、データバスオペレーションの残り部分を分担し続行する。メモリ及びI/O割当てオペレーションに関してのアドレスの長さ及びデータの長さのフォーマットも使用者によつて特定される。

P R タグピン： これらのピンは、バイダイレクショナルなピンである。出力ピンとしてのピンは、データバスオペレーション中のソースプロセッサ、即ちデータバスオペレーションを開始した単一のプロセッサ又は内部ロックステップのマスタープロセッサであるプロセッサを示す為に使用される。故に、データポートはデータバスオペレーション全期間中、データバスが送信側プロセッサに支配されていない組織において使用することもできる。例えば、メモリに脱出しの為のアドレスを送った後でR I Cはデータバスを解除することができ、メモリがデータバスを支配するのを待ち、データを送り返すことができる。データバスを一時的に

解除する理由は、他のデータバスの処理がその間行われるようにさせる為である。この解除の方針のもとにおいて、メモリからひきだしたデータは、脱出し要求のシーケンスとは異なるシーケンスにおいて戻る（これは例えば共用され、内部差込み形式のメモリ内で起こる）、P R タグ情報は、脱出しオペレーションにおいて重要である。（外部）メモリ制御器はデータがR I Cに戻ってくる時にこれらのピンをパツファしてデータに適切なタグをつける。内部的には外部データポート72又は73は割当て線168のうちの1本の上のこの情報を解読し出力して、受取ったデータを適当なメモリ位置にマップしてゆく。これらのタグピンはR I CからR I Cへの通信を行う割当てプロセッサをも指定している。

データ転送同期： これら2つのピンは、ソースと割当て先の間のデータバス上をデータが正確に転送されていることを確実にする為に必要とされる。これらのピンは、情報転送可能（I A）と情報受信（I R）である。情報転送に関するプロト

コールは、以下の通りである。データバスの制御を行いながらソースはただちにI A信号の電位を下げる。ソースはデータ/アドレスピンに情報を設定するので、I A信号は'1'まで電位が上げられる。I A信号の電位を上げることによつて全ての受信側プロセッサに情報がデータバスによつて転送中であることを知らせる。I R信号は通常低い電位にある。受信側がデータバスからの情報をパツファする時にI R信号の電位が引き上げられる。I R信号はワイヤ-ANDに接続されている。故に受信側が多重に存在する場合、I R信号は、全ての受信プロセッサが情報をパツファするまで低い電位のまま維持される。

要約すると、データバスが特定されることで融通のきく仲裁プロセスを持つ仲裁データバス回路網が構成される。情報転送の割当ては、割当てをアドレスする機能を持つので一般的なものである。情報のフォーマットはR I Cをプログラムし、メモリ及びI/Oシステムを構成することで特定されるので一般的なものである。最後に、情報の転送

は、ある点からある点の間でも又は間にいくつかの分岐を含むもの（マルチドロップ）であつてもよい。

割込みポート（ピン51-58）はこの目的を果たしている。第1の目的は、従来のマイクロコンピュータ及びマイクロプロセッサが行つていたのと同様に外部からの割込みを受けとりプロセスを行うことである。割込みの概念は、他の受信側プロセッサに割込みを送信する機能とともにこの従来の目的も達成できるよう一般化される。この一般化によつてチップ内部の通信が可能となる。チップ内部の通信を可能にするという目的は、R I Cにタスクをコーディネイトし情報を転送する役目である。チップ内部の通信システムはチップ間通信の命令（communique）の部分を送信する為に使用される。情報転送のデータ部分はメモリ間で通信される。例えば、ディスク脱出しオペレーションは、R I Cの割込みポートを使つてディスク制御器に命令を送ることによつて開始される。データ転送は、ディスクシステム及びメモリシス

テムの間の個別なデータバスにおいて行われる。

**割込みバス仲裁：** R I C の割込みポートは8本のピン51-58を有している。ピン57-58は割込みを送信中に使用される共有資源の要求を仲裁する為提供される3つのモードの仲裁が維持されている。第1のモードは、ラウンドロビン機構で仲裁される共通の割込みバスを維持している。この機構の中で「1」の信号がチップの間を循環して上記で示したように割込み資源を配分している。あるチップに於る割込み可能(1A)出力(ピン58)が隣接するチップの割込み許可(IG)入力(ピン57)に接続される。1G信号があるチップで「1」になる時このチップは割込みを発生することができる。送信すべき割込みがそのチップに存在しない場合、1A信号の電位が上昇する。第2のモードはデータポート仲裁の方針で説明したのと同じマスタースレーブモードである。通常は、マスタースレーブモードで制御している。従属R I C がマスタースレーブの1G入力信号を上昇させてアクセスを要求している場合、マスタースレーブR I C

は、好きな時にマスタースレーブの1G信号の電位を上げて選択的に従属R I C に対するバスの支配を得る。もう1つの仲裁モードは各々のチップの割込み可能及び割込み許可信号を外部仲裁ハードウェアに接続して行う。R I C が送信すべき割込みを有する時、R I C は1A信号の電位を上げる。割込みポートが割込みの要求側によつて使用可能な状態にある時、外部ハードウェアが1G信号の電位を上げる。3つの仲裁機構を用意する理由は応用に対する一般性を与える為である。ラウンドロビン仲裁機構では、少数のR I C を接続して内部通信を可能にする経済的な方法を提供している。マスタースレーブ機構では、多重な情報処理機能を持つ装置をコーディネートする為の一般的な方法の使用を可能にする。R I C を外部割込み管理回路に付加できる機能性を有することで、チップ内部通信に用いる回路網を思いのままに構成することが可能となつている。

**割込み情報：** ポートの4本のピン53-56は、外部割込み情報専用のものである。外部割込み情

報プロトコールは、最大限に使用者が規定できるように特定される部分は最小にしてある。また情報プロトコールはアドレスとデータとの区別に関して示している。また、プロトコールはメッセージの長さも示している。ここに示すものに関する特徴を以下に説明する。情報プロトコールは、あらゆる情報の第1番目の部分が割込みの受取り手を認定するアドレスであることを示している。アドレスの長さは使用者によつて指定される。割込みが送られる時は、共通な割込みバス上の全てのチップがアドレスを受けとり、それを記憶する。以下で規定されるように、割込み状況信号は情報線がアドレスを運んでいるのかデータを運んでいるのかを知らせる。状況がアドレスビットが送信されていることを示す限り、受信側のR I C はアドレス部分をバッファしておく。割当てアドレスが送信された後、各々の受信側R I C はそのアドレスを使つてチップ上のRAMメモリ66内のビットにアクセスする。N個のアドレスビットが送られる場合、高いオーダーのN-3個のビットを

使用して内部メモリのテーブル内のバイトにアドレスする。このバイトはR I C の外部割込み管理回路76に送られる。次に低いオーダーの3つのアドレスビットがこのバイトの中から1ビットを選択する。この選択されたビットが「1」である場合、このチップが割込みの割当て先であることを示す。このR I C は何か送信されてくるまで割込みと共働する残る情報をバッファしつづける。このアドレス探索機構は、任意につけ加えられる。単純な割込み機構では、受けとつたどの割込みもいずれかの受け取り手に対し送信することができる。

通信ポートの残る2本のピン51及び52は、割込みバスの状況を示す為に使用される。ピンの配置はテーブル2に示される。

テーブル2

通信状況ピンの配置

IS1      IS2  
(ピン51) (ピン52)

1	1	受信側準備
1	0	データ転送
0	1	アドレス転送
0	0	情報転送完了

2本の割込み状況線は2つの目的に使用される。第1の目的は、ソースと割込先の間非同期の和論信号を与えることである。もう1つの目的は、アドレスの転送とデータの転送を区別することである。割込み信号の転送が始まる前は、割込み信号は11の状況(即ちIS1=1及びIS2=1)である。第1のニプルをアドレスとする。第1のニプルが送られると割込み状況は01となる。全てのIS1信号がいつしよにワイヤANDで接続される。全てのIS2信号もワイヤANDで接続される。1タイムの遅延の後、割込みのソースはIS1信号をフロートの状態にする。情報がチップにロードされると各々の受信側RICは

味する。もしラウンドロビン仲裁機構が使用される場合、送信側が割込み可能信号の電位を上げて次の割込みを送信するRICの選択を開始させる。マスタスレーブオペレーションが使用され、送信側が従属RICである場合、マスターが割込み資源の制御を行う。マスターが送信側である場合、従属RICに対し割込み資源をせりあうことができることを知らせる。

通信プロセスはチップの外部割込み管理回路76にMPSが割込みを送り出すことによつて始まる。上記で示したように、優先性バイト及びランタイム情報である4ビットは、内部割込みの受信側に送られる。割込み管理回路76が内部割込みを受けとる時、優先性バイトは、テーブルエントリを示している。このテーブルはチップのRAM66内に含まれている。テーブルエントリは、メッセージブロックを示す。ランタイム情報は、メッセージブロック内のバイト数を意味する。これによつて15バイトを送信することが可能である。ランタイム情報が10進法上のゼロである場合、

IS1信号の電位をゼロに引き下げる。チップが受取った情報処理した後でこのチップはIS1信号をフロートの状態にする。全ての受取り側RICが情報のプロセスを完了すると、IS1信号は「1」を示すようになる。そして状況は11となる。送信側は送信可能な状態の次のニプルがもしあればそれを調査する。次のデータの形式は状況線によつて決定される。状況01は、アドレス情報に対応し、状況10はデータ情報に対応する。データ情報が送られると、送信側は、アドレス情報の転送に使用したのと同じプロトコルを使用する。唯一の相違点は、情報が送られる後に、送信側がIS1及びIS2両方の信号をフロートの状態にすることである。最後のニプルが送られた後で、送信側RICが状況00を出力し、割込み情報ブロックの終了したことを知らせる。遅延の後に送信側は状況信号をフロートの状態にする。このことが受信側に割込みが終了したことを知らせる。外部仲裁機構が使用される場合は、割込みの終了は次の割込みの開始が可能であることも意

メッセージブロック内の第2のバイトがメッセージ内のバイト数を有している。第2のバイトが(ランタイム情報と同時に)10進法のゼロである場合、次の2バイトがメッセージブロック等の中のバイト数を有している。

チップの外部割込み管理回路76が外部に送信する為の割込みを受けとつた場合、外部割込みバスにアクセスすることができることにより割込み管理回路76が割込み送信プロセスを開始する。仲裁機構がラウンドロビン形式である場合又は、仲裁機構がマスタスレーブ形式であつて送信側RICが従属関係にある場合、送信側は割込み許可信号が「1」になるまで待機する。マスターRICが資源を制御する時はいつでも、マスターは割込みを送ることができる。外部の仲裁機構が使用される場合、送信側RICは、割込み可能信号の電位を上げる。送信側はさらに割込み許可信号が「1」を示すようになるまで待機する。割込み許可及び割込み可能信号で制御されるハードウェアの制御方針は、開始時点で発生した2つの状

況ビットによつて決定される。これらのビットが11であればラウンドロビン仲裁機構を使用しこれらのビットが00を示せば外部仲裁機構を使用する。マスタースレーブ仲裁機構は10を示す場合に使用される。

外部割込みバスを支配できるようになつた後、外部割込み管理回路が割当てアドレスを送る。割込み管理回路76はチップのRAM66からデータを取りだしその1タイムに1ニプルの情報を割込みポートの情報線53-56に送る。ソースプロセッサが外部割込み管理回路76に優先性バイトを送り、外部割込み管理回路76の送信オペレーションを開始させることによつて、メッセージブロックの開始が知らされる。優先性バイトは、外部割込みテーブル内のエントリを示す。この外部割込みテーブルは、割込みが外部的にも内部的にも両方に対し送信されるのかどうかを示すバイト及びメッセージブロックの開始アドレスを示すバイトの両方のバイトを含んでいる。テーブル内の第1のバイトは、自己に対する割込み、即ち外

部及び内部に対し回帰的に送信される割込みを示す為に使用される。この型の割込みは以下で示すような、外部ロックステップに対する割込みに必要とされる。このバイトの一番左のニプルは全て「1」である。右のニプルは、内部のどのプロセッサに割込みを行うかを示す為に使用される。ニプルの最上位ビットが、PR3に対応し、次の最上位ビットは、PR2に対応し、以下この通り対応する。もし、右側ニプルのいずれかのビットに「1」が含まれる場合、それに対応するプロセッサは、そのプロセッサに対する割込みを受けとっている。割当てアドレスの最後はゼロのバイトで位置を示される。外部割込み管理76がゼロのバイトの存在を検索する時、もしゼロが存在すれば割込みメッセージのデータ部分を送る。データ部分の終了は、またゼロのバイトで知らされる。次に割込み管理回路76はゼロを検知した場合、アドレスのソースを送る。内部割込みデータ構造は、第35図に示される。外部割込み管理回路に知らされたメッセージブロックの長さによつて決定さ

れた通りのメッセージブロックが送り終わるまで割込み管理回路がニプルを送る。メッセージブロックが送られた後で且つ外の内部及び外部に対する割込みのプロセスが始まる前に、外部割込み管理回路76はテーブル内に割込み状況を割込みを発生したプロセッサに対し割込みを送る。このプロセッサが割込みを開始する。また、割込みメッセージブロックの送信が例外的に中断された場合もこの状況が同じテーブル内に書き込まれ発生元であるプロセッサに割込みが行われる。この割込み送信プロセスは第36図に示す。

上記で示したように、共通割込みバス上の全てのRICは割当てアドレスを受けとり、これをそれぞれの持つ外部割込み管理回路76内にバッファする。アドレスを受取つた後で、各々の外部割込み管理回路76がひとつのビットにアクセスし、そのあるそれぞれのチップが割込みの割合で先であるかを判断する。あるRICが割込みの割当て先である場合、外部割込み管理回路76は残るメッセージをそのチップの内部RAM内にバッファ

する。メッセージブロックのデータ部分の終了までくると、割込み管理回路はゼロのバイトを書きこみデータ部分の終了を示す。割込み管理回路は「1」が送られてきた場合割込みのソースアドレスもバッファする。メッセージブロックを送りおえた後、割当て先RICの外部割込み管理回路76は、割当てアドレスを使用して内部RAM66内の外部割込みマップテーブルにアクセスし、これによつてRIC上のどのプロセッサが割込みを受けとるのか又は割込みは内部的にどの程度の優先性を持つのかを判断している。内部割込みの優先性は、メッセージブロックの開始位置及びメッセージの長さが書きこまれる外部割込みメッセージテーブル内の位置を指定する為に使用される。さらに、外部割込み管理回路は割込みをチップ上のプロセッサに送る。割込み管理回路は外部割込みメッセージテーブル内の情報を使つてメッセージブロックにアクセスしこのプロセスを行う。

外部割込み管理回路が使用するRAM領域のメモリ管理は、割込みを送つたり受けたりするプロ

セツサーにより行われる。プロセッサが外部割込みを送る時、割込みはポインタをメッセージブロックにわたしている。外部割込み管理回路76が割込みを送った後に、送信したプロセッサには送った割込みの状況が知らされる。割込みが誤りなく送られた場合、プロセッサは改めてメッセージブロックのメモリ領域の使用を請求できる。外部割込み管理回路76は、プロセッサPR0~PR3に自分自身のメモリの管理を行わせる機能はない。故に外部割込み管理回路76は、外部から受けとったメッセージブロックを書きこむ為の領域を確保する為にメモリの管理を必要とする。外部割込み管理回路はメッセージブロックを記憶しておく為に2つのメモリポインタ：現在のメッセージブロックポインタ(CMB)及び次のメッセージブロックポインタ(NMB)を有している。CMBが有効であれば外部割込み管理回路76はこのポインタをメッセージブロックの開始として使用し、受取ったメッセージブロックの各々のバイトが書きこまれた後にアドレスポインタをインクリメント

とする。割込みメッセージブロックを受けとった後、割当てプロセッサは、割込みを知らされる。外部割込み管理回路はNMBを直ちにCBMに変えて、NMBが有効でないことを記録する。割込みを受けとったプロセッサはただちに新しいメッセージのブロック領域を計算して出すという高い優先順位のプロセスを開始する。この新しいメッセージブロックの開始アドレスは外部メモリ管理回路のNMBレジスタに書きこまれる。外部メモリ管理回路がCBM及びNMBのどちらのポインタも有効でないという状態を示すようになると、割込みの送信側に「用意できていない」という信号を送って、どちらかのポインタが有効となるまでメッセージブロックの送信は停止する。

外部割込み管理76は開始時点で初期化される最長ブロックパラメータを有している。最大の長さより長いメッセージブロックが送信される場合外部割込み管理回路は「受け取り準備ができていない」という信号を割込みの送信側に送る。また外部割込み管理回路はソースプロセッサを判断し

それらにメッセージブロックのあふれを意味する割込みを送る。プロセッサはプログラムされていた動作を実行する。例えば送信側が時間外に超過した後で、プロセッサは割込みを途中で停止させることはできなくすることもできる。この他にも、外部割込み管理回路76に他のメッセージブロックポインタを送ることもできる。外部割込み管理回路76は内部割込みによつて新しい、メモリポインタで開始するメッセージブロックをひき継ぎ受けとらなくてはならないことを知らされる。この後者の方法は、メッセージブロックの長さが様々で、最長のブロックの長さが大多数のメッセージブロックより長い、又は等しくセットされる場合に使用される。

この割込みインターフェースを提供する目的は、使用者が割込み機構を規定できるようにスペクトルを提供する為である。最も簡単な割込み機構は従来のマイクロプロセッサで使用する割込み機構と類似している。このより簡単な機構より更に融通性を拡大する為には内部的にプログラム構造

を形成する必要がある。

状況ポート1(ピン59-67)及び状況ポート2(ピン68-76)である2つの状況ポートが提供される。状況ポートの信号は同一である。故に状況ポート1のみに関し説明を行う。状況ポートの主要な機能の1つは、異なるRIC上のロックステップモードの2つのプロセッサに信号を提供することである。状況ポート1は、PR0又はPR1、又はロックステップで接続されるPR0及びPR1を外部プロセッサにロックステップで接続する為に使用することができる。状況ポート2は、PR2又はPR3に関し、あるいは、PR2又はPR3がロックステップ内で最も重要なマスタープロセッサである内部のロックステップに関し、外部的にロックステップを形成する為に使用することができる。各々の状況ポートには、4つの型のピン機能がある：即ち、ALUの結果状況、桁上げ連結、シフト/循環連結及びチップ内部の同期である。状況ポート1(ピン59-67)は、内部状況バス52及び54に外部状況ポート75

(第1図を参照せよ)を通して接続される。同様にして状況ポート2は、外部状況ポート74(第1及び第2図で示す)を通して接続する。

**ALUの結果状況：** ALUの結果を示す為には4つの線59-62が存在する。負の結果状況N、ゼロの結果状況Z、桁の結果状況C、及びオーバーフローの結果状況Vという信号が存在する。これら4つの信号は、各々外部的にロックステップで結ばれる状況ポート全てが接続する外部状況バスにワイヤーAND接続される。これらの信号がコード化され16のALU結果状況のうち1つを示すようになる。これらの線のオペレーションは上記で示したような内部ロックステップ構成におけるALU結果状況の内部での発生及び(内部状況バス52及び54における)ALUの接続のオペレーションとほとんど同様である。

**桁上げ連結：** 桁上げ連結(carry linkage)は桁上げ信号(ピン63)及び桁下げ信号(ピン64)を有している。あるRICの桁下げ信号は次に高順位のRICの桁上げ信号と接続される。

ステップからはずれることができる。故にマイクロ命令をとりだす時間は、ロックステップで接続されるプロセッサを有するRICの間で様々である。チップ間の同期ピンはフラグとしての役めをし、各々のプロセッサが先のマイクロ命令を完了し次のマイクロ命令を取り出したことを知らせており次のマイクロ命令を実行する用意ができていることを示す。チップ間の同期ピンはワイヤーAND接続される。全ての外部的にロックステップで接続されるプロセッサの次のマイクロ命令を実行する用意ができた時、チップ間の同期線の電位が上がる。もし1つ以上のPRの用意ができていない場合、線の電位は低くなる。同期線の電位が高い時、次のクロックサイクルで実行が開始する。(共通な外部ロックステップ内のPRを持つ全てのRICは同じシステムクロックを使用しなくてはならない。)実行が始まった少し後で、チップ内部同期線の電位は全てのPRが次のマイクロ命令を実行する用意のできた状態になるまで低くなっている。PRが外部的にロックステップで接続

あるRICで完了した演算が使用される場合、最上位プロセッサは最下位プロセッサの桁上げ信号に接続される。

**シフト/循環連結：** シフト/循環連結(shift/rotate linkage)は、外部的にロックステップで接続するプロセッサ間のシフトオペレーションを実行する為に使用される。RICのシフト/循環高電位信号(ピン65)は、次に最上位のRICのシフト/循環低電位信号(ピン66)に接続される。最上位RICのシフト/循環高電位信号は、最下位RICのシフト/循環低電位信号に接続されて循環連結が形成されている。

**チップ間の同期：** チップ間の同期(ピン76)を行う為には、外部的にロックステップで接続するプロセッサが確実にフェイズ内で同じ命令を実行するようにしなくてはならない。外部ロックステップ内のプロセッサは、同じRICの中の1つのRICの上に形成される他のプロセッサが外部のロックステップで接続するプロセッサとは独立して作動されることから、同期を行わずにロック

される時、マイクロ命令の実行中はPRに対する割込みはおこつてはならない。この割込みの制限によつて、マイクロ命令の実行が開始した後でも外部でロックステップ接続するプロセッサ間の同期は確実に保たれる。この制限で得る利点は、状況ポートのピンの為に他の同期手段を設ける必要がないという点である。外部でロックステップで接続するプロセッサに関してはもう1つの事項がある。外部のロックステップ内のプロセッサに対する割込みは、プロセッサの同期を妨害することになるので、割込みによる効果を考慮しなくてはならない。追加のピンあるいは追加のオーバーヘッドを使わずに外部的なロックステップ接続の同期を維持する為には、外部ロックステップに対する割込みは、外部ロックステップ内の全てのMSPに対して行われるよう制限されなくてはならない。割込みのサブセットに割込みを実行することは可能なのでこれはあまり厳格な制限ではない。外部ロックステップ全体に対し割込みを送ることを必要とするこの制限によつて、外部ロックステップ



内のちようどプロセスを開始しようとしているプロセッサに割込みの名前を知らせる必要性を省くことができる。外部ロックステップ全体に送られる割込みは適切に実行される。これは、割込みが行われる外部ロックステップを有する全てのR ICが同時に同じ割込みを受けとることによる。R ICプロセスの外部割込み管理回路は、受取ったオーダーで割込みを行う。故に、外部ロックステップ内の全てのプロセッサは同じオーダーで割込みを受取る。R ICの自己に対する割込み( self interrupt ) に関しての特徴は、外部ロックステップに対する割込みが割込み先である外部ロックステップ内のプロセッサを有するR IC内で発生される場合に必要となる。前に述べたように、自己に対する割込みは、割込みの送信側にとってその送信側自身が受信も行っているものとして取り扱われる。

上記で説明した通り、外部ロックステップ内のプロセッサはマイクロ命令の実行中は割込みが行われることはない。マイクロ命令が終了すると割

込を発生することができる。例えば、R ICに使用されるC R T 制御回路の場合、N 信号は内部的に制御されて水平同期信号を発生する。また、チップ間同期信号は内部制御システム( 即ちD L A 2 2 等 ) に対する直接の割込みの時に使用される。例えば、チップ間同期信号は、内部的には、外部クロックとして通訳される。この外部クロックは、各クロックサイクルの期間に所定のオペレーションを発生させる。

外部ロックステップの全てのプロセスは、割込みによつて開始される。R ICのリセットによつて外部ロックステップ内のマスタープロセッサが自己に対する割込みを発生し、外部ロックステップを開始させる。プロセスの完成時点で、マスタープロセッサは自己に対する割込みを発生し、最も高い優先順位を持つ別々のプロセスを開始する。

リセットインR I ( ピン7 7 ) 及びリセットアウトR O ( ピン7 8 ) の2本のチップ制御線が提供されている。全てのR ICから接続するR I 及びR O 信号は別々に接続される。R I 信号はアク

込みが行われたプロセッサは4本のA L U 結果状況線にコードを発生する。( 即ち、0 1 1 1 である。 ) このコードは、外部ロックステップの割込みが少なくとも1つのプロセッサによつて受けとられたことを示している。これによつて外部ロックステップの実行は停止する。各々のプロセッサはコンテキストスイッチが実行されるとチップ間M S P 同期線の電位を低く保つのをやめる。現在の割込みより高い優先順位を持つ他の割込みが、現在の割込みが入れ変わろうとしている期間中に外部ロックステップに送られる場合、全てのプロセッサが現在の割込みに変わるまで次の割込みは無視される。

状況ポートにはもう1つ重要な機能がある。外部A L U 状況ピンN Z C V は、制御D L A 2 2 への入力として実際の内部A L U 状況ピンを配置することによつて決定される値を持つ。制御D L A 2 2 への入力は外部ピンに送られた値を発生する。故に上記で示したような機能によつて、リアルタイム信号を発生する為に使用される実際の外部信

タイプで高い電位である。R I 信号が「1」を示すまで電位を上昇されるとR ICは自分自身を初期化を開始しオペレーションに備える。R O 信号は、有効にワイヤーA N D で接続されている。R ICが初期化オペレーションを完了した時、電位を低くしてあるR O 信号はフロートの状態になる。全てのR ICが初期化を完了すると、R O 信号は高い電位となりシステムが初期化を完了したことを示す。

チップが初期化されると、プロセッサP R 3 はただちに最も高い優先順位2 5 5を持つプロセスを開始させる。この場合プロセッサP R 3 がマスターである。プロセッサP R 3 は次にチップ上のR A M 6.6 の位置「0」に固定アドレスをロードし、プロセッサP R 3 はそこで他のプロセッサに対する割込みを行いこれに必要なプロセスを開始する。

好ましい実施例において、この中のR ICは、高出力部分に使用されるC M O Sを持つ1マイクロメーターの微細加工によるN M O Sに実施され

ている。2つの出力ピン81及び82が3ゲルトと接地電位で使用される。

更に、本実施例はR1C上にクロック発生器を設置し、2つのクロック入力(ピン79及び80)の間にのみ結晶を設置する必要がある。その代わりに、所望の通り例えば多重R1Cを使つたシステムに実施する場合では、外部クロック回路をこれらのピンと接続することができる。

第37図から第41図は、1つ又は2つ以上のR1Cチップを使つて与えられたある特定の構成を示す。第37図は、16ビットパイプラインR1C構成を示す。パイプラインプロセスは4段のプロセスにのみ限られるわけではなく、追加のR1Cをいつしよに連結することで望むだけの数の段を含むことができることに注意してほしい。

第38図は、32ビットパイプラインを示している。プロセッサPR1及びPR2は、プロセッサPR1及びPR0と同様ロックステップで接続されることに注意して欲しい。また、2段のパイプラインのみが示されているが、単に十分な数の

R1Cチップを連結するだけでパイプラインの段をいくつでも望みの数にすることができることも注意していなくてはならない。

第39図は、64ビットの外部ロックステップパイプライン構成を示している。R1CAのプロセッサPR3及びPR2及びR1CBのプロセッサPR3及びPR2は、全てのプロセッサPR1及びPR0と同様にロックステップで接続される。第39図において、両方のR1Cから接続する32ビット出力は、それぞれの単一の16ビットデータポートにおいて、多重化されているものとして示している。これはパイプライン構成の中の1つのデータポートは、前の段から与えられたデータを受け取る為に必要とされるからである。異なるチップ上のプロセッサPR3及びPR2を含むように外部ロックステップ構造の大きさを拡大することによつて、より大きなパイプラインを作ることさえ可能である。故にパイプラインワードの規模は、32ビットインクレメントまで拡大することができる。32ビットのインクレメントは、

桁上げ/桁下げ信号及びシフト/循環高/低信号を除き状況ポートの全てのピンを接続して特定のステージを作り出すことによつてつけ加えられる。(桁及びシフト/循環信号以外の)これらの信号は、ワイヤ-ANDでいつしよに接続される時適正に機能する。桁下げ信号はピンの中の次に最高位の桁を示すものに接続される。シフト/循環高電位ピンは次に最高位シフト/循環低電位ピンに接続する。(最高位シフト/循環高電位ピンは、最下位シフト循環低電位ピンに接続する。)

第40図は、64ビットの規模のステージのパイプラインを示す。このステージは、最初に再構成可能なIC上の4つ全てのプロセッサをロックステップで接続することによつて形成される。第40図では、別々の16ビットの入力及び出力ポートとして示されているので、4つのフェイズでの多重なデータ転送が必要となる。各々のICの2つのデータポートを統合することも可能であるので32ビットの並列ポートが形成される。これによつて第40図の64ビットパイプラインのステ

ージ構成は、4クロックサイクルよりむしろ2クロックサイクルでデータ転送を実行している。しかしながらこれによつて、32ビットの転送を可能にし多重ステージパイプラインのステージの他のペアとの間を並列にする別々の絶縁されたスイッチの組が必要になる。故に、多重16ビットポートを使用した方がはるかに簡単であり、一般的に好ましい。

第41図は128ビットの大規模ハイブリッドロックステッププロセス実行素子を形成するように接続された2つのR1Cチップを示している。内部的には各々のチップの上の4つのプロセッサはロックステップで接続されている。各々のチップ上に1つだけの状況ポートが使用されて図で示す通り2つのチップ間にロックステップを形成しているので、いずれかの多重化法によつて64ビットの規模にした同様のハイブリッドプロセス実行素子を構成することが可能である。

多重R1C構成において、割込み操作通信網を使う樹系図を用いることでのその融通性及び複雑

性をさらに増加することができる。

以下の3種の割込みは、一定の集積形式で取り扱われる。(1)チップ内部割込み：これらの割込みのソース及び割当てプロセッサは同一R I C上に存在するのでこれらの割込みが1つのR I Cチップ上で発生し完全に同一チップ上で処理される。(2)チップ間割込み：これらの割込みのソースプロセッサ及び割当てプロセッサが異なるチップ上に存在している。(3)I/O割込み：これらの割込みはI/O装置で発生し、1つ又は2つ以上のR I Cチップ上に存在するプロセッサに対し行われる。

チップ内部割込みは通常チップ上でダイナミックな再構成を行う為に使用される。例えば、内部的ロックステップモード又はパイプラインモードのオペレーションを実行させる為に使用される。内部的な割込みは、同じR I Cチップ上の他のプロセッサからサービスを受ける為にも使用される。チップ内部割込みは多重チップ構成に於て重大な役目を果たしており、チップ外又はチップ内のロックステップパイプラインモードのような様々な

多重モードを開始させる為に使用している。チップ間割込みも、異なるチップ上に異なるプロセッサが使用される時に、多重プロセッサの群様におけるプロセッサ間の通信を行う為に使用される。一般的を例は、異なるチップ上に形成された中央プロセッサ及びI/Oプロセッサを含む多重チップ構成である。中央プロセッサ及びI/Oプロセッサは割込みを介して通信を行つている。I/Oプロセッサは、中央プロセッサと共通バスを共有していて、中央プロセッサのメモリとI/O装置の間で情報の転送を行つている。

故に、チップ上のどのプロセッサも3種類のソースから割込みを受けとる可能性がある。：同一チップ上の他のプロセッサ、又は異なるチップ上のどれかのプロセッサ又はI/O装置からの割込みである。以下の集積割込み機構は一定の型式でこれらの割込みを処理している。いずれかの割込みに応答して、割当てプロセッサの割込み管理回路12は対応するスケジューラ16に指示して割込みの優先順位及び現在実行中のプロセスの優先順

位に従つて適当なプロセスのスケジュールを行う。

上記で示した様に、各々のR I Cは外部割込み管理回路76を有し、この割込み管理回路が外部ソースから受取つた割込みの受け取り、記憶及び内部プロセッサへの通知を管理していて、内部プロセッサによつて発生された外部割当て先に対する割込みの送信も管理している。

多数のR I Cチップを含むことのできるシステムの為の上記のような割込み管理機構を一般化する為に、我々は「クラスター(cluster)」の概念を導入している。クラスターとは、相互に割込みの送信が可能な一組の例えば4つのチップから成る組のことである。第42図は、4つのチップを持つクラスターの例を示している。故に、割込みの階級的組織(hierarchical organization)においては、クラスターはチップより1つ上位のレベルである。本実施例の割込みシステムでは、クラスター内部のチップの役目は、チップ内におけるプロセッサの役目と類似する。ここで我々はバスを結合することでこのクラスター内に発生す

るチップ間割込み、即ちクラスター内部割込みを容易に操作することができる。このようなバスは「クラスター割込みバス」と呼ばれ、第42図に示されている。このバスはまたクラスター内部で発生し、クラスター外部に割当て先を持つクラスター即ちクラスター間割込みを運搬している。同様にバスはクラスター外部で発生し、クラスター内のチップの内の1つに於るあるプロセッサに割当てるI/O及びクラスター間割込みの運搬に使用されている。この階級的機構をもつと有効に利用する為には、割込み管理回路218が加えられる。この割込み管理は外部割込み管理回路76がR I Cチップに対し行つていたと同じ役目をクラスターに対し行つている。故に各々のチップ216の外部割込み管理回路76は、クラスター割込みバス214及びクラスター割込み管理回路218とインターフェースしていなくてはならない。

クラスター割込み管理回路218の第1の機能は、クラスター割込みバス214とクラスター間及びI/Oの割込みの間のインターフェースとし

て動くことである。故にクラスター内で発生したクラスター間割込みは、クラスター割込みバス214及びクラスター割込み管理回路218を介し外部クラスターに送られる。同様に、クラスター外部で発生したがクラスター内にあるチップを割合て先にするクラスター間及び1/0の割込みは、クラスター割込み管理回路218及びクラスター割込みバス214を介し割当てチップに送られる。クラスター割込み管理回路218及びクラスター内にある(例えば)4つのチップはクラスター割込みバス214を共用している。クラスター割込みバス214に関する仲裁方針は、ラウンドロビンに限られない。R1C割込みインターフェース76によつて外部的な制御方法によつて仲裁を行うことが可能である。故に違う仲裁機構を必要とするクラスターは、それぞれの持つ各々のクラスター割込み管理回路218内にその機構を設けている。クラスター割込み管理回路218は優先順位又は位置に基づくような仲裁方法を使用することもできる。優先順位に基づく仲裁方針を

とつた場合、クラスター割込みバスを共用する全ての競争者(即ち、チップ216及びクラスター割込み管理回路218は、チップ外部に存在するバス仲裁論理に(他のチップに送る為の)割込みの優先順位の判断をゆだねている。そこで仲裁論理が最も高い優先順位を持つ競争中の割込みソースを決定しバスの制御をまかせる。位置に基づく仲裁方針が使用される場合、バスの使用権につき紛争が起きた場合、クラスター内部の競争者の位置が誰にバスの支配を与えるかを決定する。例えば第42図では、位置に基づく方針を使つて決定する場合、争いが起つたならばクラスター割込み管理218が常に第1の優先順位を持つとすることができ、それぞれチップ0、1、2、3が続く。設計者は、考想中の応用に適合するようにもつと複雑な仲裁方針を自由に選択することができる。しかし、優先順位に基づくようなより複雑な方針にはチップ外部にかなりの論理を必要とするが、ラウンドロビン又は位置に基づくような簡単な方針はチップ外部に非常にわずかな論理を用意する

だけで使用できる。

クラスター割込み管理回路218の第2の機能は、クラスター外部に出ていくクラスター間割込みの為の、またクラスター内に入ってくるクラスター間及び1/0割込みの為のバッファとして働くことである。このバッファする機能は、本質的にクラスター内部の割込み運送とクラスター外部の割込み運送を隔離している。クラスター割込み管理回路218はクラスター内部外部両側の割込み運送を同時に操作するインターフェースとして設計されている。どのようなクラスター管理回路はR1C又はR1C内の1つ又は2つのプロセッサに設けることができる。

この点で生まれる明らかな疑問点としては、2つ以上のクラスターを持つシステムでいかにして管理を行うかということである。これらのクラスターは、自分のクラスター割込みバス及びクラスター割込み管理回路を介しクラスター間割込みを送っている。さらに階級的組織による管理機構を発展させて、一組の(例えば4つの)クラスター

を「マクロクラスター」と呼ぶことにする。マクロクラスター222内のクラスター220は、「マクロクラスター割込み管理回路」226に接続する「マクロクラスター割込みバス」224を共有している。マクロクラスター割込みバス224及びマクロクラスター割込み管理回路226はマクロクラスター222内に於て、クラスター割込みバス214及びクラスター割込み管理回路218がクラスター内部で行つていたのと同じ役目を果たしている。我々は、この考え方をさらに発展させることができる。故に(例えば)4つのマクロクラスター222で次に高いレベルになるような実在を作りだすと考えることができさらにそれ以上発展させることも可能である。最後に、好きな数だけレベルを設けた後で割込みを転送するバスを共有する(例えば)4つのサブシステム234まで拡張し、これらから成るシステムの段階まで送ることができる。このバスはシステム割込みバス232と呼ぶ。

第43図は、サブシステム0及び1である2つ

のサブシステム234から成るシステムを示している。サブシステム1は、マクロクラスター0及び1を含んでいてサブシステム0は、ただ1つのマクロクラスター即ちマクロクラスター0を含んでいる。サブシステム1のマクロクラスター0は4つのクラスターから成り、サブシステム1のマクロクラスター1は2つのクラスターから成りサブシステム0のマクロクラスターは2つのクラスターから成る。各々のクラスターは4つのチップを有している。この例ではサブシステム1は1つだけのマクロクラスターを有している。故にこのマクロクラスターの割込み管理機能はサブシステム0の割込み管理に委託することができる。これによつてサブシステム0ではマクロクラスター割込み管理回路及びマクロクラスター割込みバスを除くことができる。ここでは、割込みの階級的構造の考え方を示している。

第44図では、割込みの階級的構造の考え方を示す樹系図を用いて第43図のシステムが示されている。RICチップ216内のプロセッサPRO

ップ76、クラスター218、マクロクラスター226……サブシステム230の割込み管理回路は次々とより高くなる各々のレベルに存在する。これらに1、2……nと番号がついている。プロセッサは、その完全な「アドレス」を与えることによつて即ちサブシステム……、マクロクラスター、クラスター、チップ及びプロセッサを規定することによつてプロセッサを完全に規定することができる。故に各々のアドレスは、サブシステム……マクロクラスター、クラスター、チップ及びプロセッサを区別する為の構成部を有している。我々は、このアドレス構成部に同様に階級的組織のレベルを与えることができる。プロセッサPRO-PR3を規定するアドレス構成部は、最下位階級即ちレベル0とする。チップ、クラスター、マクロクラスター、……サブシステムを特定するアドレス構成部が次々と高くなるレベルにそれぞれ属する。これらはレベル1、2……nと名づけられている。

チップ内のプロセッサ間割込みは、最も頻繁に

-PR3は樹系図の最下位階級に存在し、木にたとえるなら「葉」の部分にあたる。次に高い段階にはRICチップ216を示す節がある。その次の2つの段階にある節はクラスター220、その上はマクロクラスター222を示す。最後に樹系図の根元では完全なシステム全体を示している。

樹系図にはレベルに番号をつけることができる。プロセッサはレベル0とする、引き続きより高いレベルに存在するチップ216、クラスター220マクロクラスター222は、それぞれレベル1、2、3……nで示すことができる。同様にして割込みの階級的組織におけるレベルにも番号をつけることができる。割込みバスに関しては、チップ割込みバス88はレベル0と番号のついた最下位レベルに存在する。クラスター214、マクロクラスター224、……サブシステム228及びシステム割込みバスは次々と高くなる各レベルに存在する。これらはそれぞれ1、2……nと番号がつけられる。プロセッサの割込み管理回路12は最下位レベル即ちレベル0に存在する。チ

起こる割込みであると予想されるのでこれらの割込みの通信及び処理は一般的にできるだけ迅速に完了しなくてはならない。上記に示した通り、1つの割込みについてのソース、割当て、優先順位及び関連ランタイムに関する情報は、一度ソースプロセッサがチップ割込みバス88の支配を握るとちょうど1バスサイクル内で通信される。クラスター内でのチップ間割込みは2番目に一般的な割込みであると予想される。本発明の割込み組織では、クラスター割込みバス214を使つてソースプロセッサからのソースチップ216及びプロセッサPRO-PR3、割合でチップ及びプロセッサ優先性及びランタイムを確定する情報をクラスター割込み管理回路218に8バスサイクルで通信することが可能となつている。マクロクラスター内のクラスター間割込みはその次に最も一般的な割込みであると予想される。本発明の割込み組織では、ソースプロセッサからの割込みに関してのソースクラスター、チップ、プロセッサ、および割合でクラスターチップ、プロセッサおよび

優先性及びランタイムを確定する情報は、マクロクラスター割込みパスを用いて10パスサイクルでマクロクラスター割込み管理回路226に通信可能となつてゐる。次々と高くなつてゆく各段階における割込み期間での同様の情報の通信には、レベルの高くなるごとに2パスサイクルのみ追加した期間が必要となる。

第44図は、3つのサンプル割込み1、2、及び3の期間中の情報の流れる方向も示している。この情報は、割込みのソースと割当て先、優先順位及びランタイム情報を規定している。我々はこの情報を「割込み情報」と呼ぶ。割込み1は、サブシステム1の中のマクロクラスター1内のクラスター3のチップ2に属するPR3で発生し、同一チップ内のPR0に割当てられている。割込み2は、サブシステム1の中のマクロクラスター0内のクラスター1のチップ3に属するPR3で発生し、ソースプロセッサがあるのと同じマクロクラスター内のクラスター0のチップ0上のPR0、PR1、PR2及びPR3に割当てられている。

スを紹介レベル(1-1)の他の割込み管理回路に移動するか、のいずれかである。例えば、第44図の割込み3は割込み情報がサブシステム1内を移動している間は上昇フェイズである。割込み情報がサブシステム1の割込み管理回路からシステム割込みパスでサブシステム0の割込み管理回路に移動する時、非上昇フェイズが始まる。情報がサブシステム0内を移動する間、情報が割当て先プロセッサに到達するまで非上昇フェイズが継続する。

上記の割込み3のように、ソース及び割当て先プロセッサが異なるサブシステム内に存在する場合、割込み情報は最長距離を移動し、最長の時間を要する。一方、割込み1のようにソース及び割当て先プロセッサが同一チップ上に存在する場合、距離及び時間は最短になる。故に、一番あまり発生しない割込みが通信又はプロセスに存在する中で最長の時間を要する一方最も頻繁に発生する割込みは最短時間で通信及びプロセスが行われる。この望ましい利点が割込みの階級的組織から直接的に

割込み3は、サブシステム0のマクロクラスター内のクラスター0に属するチップ1、2、及び3上のPR0で発生する。割込み情報は一番下の「葉」から開始し、必要な限り上まで移動してゆく。即ちレベルを上へ上つてゆき、次に系図を下まで下つて割当て先プロセッサを示す「葉」の部分まで達する。割込み情報が樹系図内を移動する時間内における割込みの移動を我々は、2つのフェイズに区別している。これらのフェイズは、「上昇」フェイズ及び「非上昇」フェイズと呼ばれる。上昇フェイズの期間中、割込み情報はきちんと樹系図を上つてゆく、即ち、情報がレベル1の割込み管理回路からレベル1の割込みパスを紹介レベル(1+1)の他の割込み管理回路に移動する。非上昇フェイズでは、割込み情報は、系図の同じレベルを移動する、即ちレベル1の割込み管理回路からレベル1の割込みパスを紹介レベル1の他の割込み管理回路に移動するか、あるいはまた系図の下に向つておりてゆく即ちレベル1の割込み管理回路からレベル(1-1)の割込みパ

生じる。

一般的なプロトコールはここでは、1番目のレベル ( $1 \leq i \leq n$ ;  $i=0$ である場合はRIC外部割込み管理回路を意味する)の割込みバス及び割込み管理回路であるとの見地で説明される。第45図で示すように、段階1の割込みバスは、レベル1の例えば4つまでの割込み管理回路とレベル ( $i+1$ )の1つ割込み管理回路によつて共有されることができる。一方、レベル1の割込み管理回路がレベル ( $i-1$ )の割込みバスとインターフェースし、レベル ( $i+1$ )の割込み管理回路がレベル ( $i+1$ )の割込みバスとインターフェースしている。我々は、ここでレベル1の割込み管理回路をとりあげて、割込みの上昇及び非上昇フェイズの区別を再考してみる。

割込みの上昇フェイズの間、割込みはレベル1の割込み管理回路から送られる。割込みは、レベル1の割込みバスを介し割込みの階級的組織の樹系図を上に向つてゆきレベル ( $i+1$ )の割込み管理回路に達する。上昇フェイズ期間中の割込みの移動は、第45図に示されている。

当て先として指定する。レベル ( $i+1$ )の割込み管理回路は、自分がこの割込みの割当て先であるかを判断する。各々の割込みの長さは、ソース及び割当てアドレスの長さによつて決まる。レベル ( $i+1$ )の割込み管理回路が最も長い場合の割込みも記憶するのに十分なバッファスペースを有している場合、割込み管理回路は割込み情報の受取り及び記憶を開始する。さもなければ、割込み管理回路は、割込み情報の最初のニプルを受けとつた事を知らせない。第46図では、転送同期信号は、割込み管理回路が割込み情報を記憶する為に十分なバッファスペースを有していることを意味する最初のニプルが受けとられたということを示している。

バスサイクル1から ( $2n+4$ )まで: これらのサイクルは、割込みの割当てアドレス、割込みの優先順位、及び割込みのソースアドレスを指定する情報を転送する為に使用される。割当てアドレスは、以下の順で送られる。; サブシステム、..... マクロクラスター、クラスター、チップ及び

割込みの非上昇フェイズの間、割込みは、レベル1の割込み管理回路によつて送り出され、同じレベルに留まる。即ちレベル1の割込みバスを移動し、レベル ( $i+1$ )の管理回路に移動するか代わりにレベル1の他の割込み管理回路にゆくか又は、レベル ( $i-1$ )の割込みバスを介し割込みの階級的組織の樹系図を下に向つて移動しレベル ( $i-1$ )の割込み管理回路に達するかのいずれかで移動する。割込み移動の非上昇フェイズの期間中の割込みの動きも第45図に示されている。

割込みの上昇フェイズ中に使用するプロトコールが第46図に示される。特に、レベル1の割込み管理回路2は、割込みをレベル1の割込みバスを介しレベル ( $i+1$ )の割込み管理回路に送りだしている。この割込みの期間中以下の動作が行われる。まずレベル1の管理回路2がレベル1の割込みバスを制御する。

バスサイクル0について: レベル1のソース割込み管理が割込み情報線に全てゼロを送つてレベル ( $i+1$ )の割込み管理回路をこの割込みの割

PR即ち、関連するアドレス構成要素の最高のもので開始して、減少オーダーで階級組織内を下に向つてプロセスが進行し、PRの認定まで至っている。第46図では、サブシステム0、マクロクラスター1、クラスター3、チップ1及び2、プロセッサPR2及びPR3が割当て先として特定されている。割込みの優先順位は127であつてランタイム情報は34である。割込みのソースアドレスはサブシステム2、マクロクラスター1、クラスター2、チップ3PR0である。ここで関連する最高レベルのアドレスがサブシステムアドレスであると仮定すると、最悪の場合でもソース及び割当て先を認定する情報を転送する為には2nサイクルが必要とされる。割込みの優先順位及びその他関連情報を送信する為には、あと4サイクルが必要とされる。例えば、最高レベルの関連アドレスがマクロクラスターまでだけである場合、マクロクラスター、クラスター、チップ及びプロセッサだけのシーケンスがバスサイクル1から5までで特定される必要がある。バスサイクル

6及び7は、割込みの優先順位を転送する為に使用され、バスサイクル8及び9は、割込みについてのランタイム情報を転送する為に使用される。ソースアドレスはバスサイクル10から14で送られる。

上昇プロトコールと非上昇プロトコールの間には2つの違いがある。1つは、上昇プロトコールでは、割込みが次に高いレベルへと送られることである。非上昇プロトコールでは、割込みは次に低いレベルへ送られるか又は、同じレベルの割当て先に送られる。2つのプロトコールの間でのもう一つの違いは、ソース及び割当てアドレスが送られる順序に関してである。上昇プロトコールに関しては、最も高いレベルのアドレスが第1に送られる。これに対し非上昇プロトコールでは階級的組織の中で次のレベルがアドレスされて送られている。

以下の例でこれらのプロトコールをより具体的に説明する。サブシステム、マクロクラスター、チップ及びプロセッサPRから成る階級的組織を

持つシステムを考えてみる。あるサブシステムで発生しどれか他のサブシステムに移動するよう割当てられた割込みは、テーブル3で示すように階級的組織の中で移動する。テーブルは各々のレベルでの(我々はチップ外の割込みプロトコールについて説明を行っているのでレベル1から始まっている)割込みデータバス上の割込みの運行を示しており、ソース及び割込み先を認定する情報が転送されるシーケンスを示している。

テーブル3は、以下の点につき示している。割込み非上昇フェイズが開始することは、割込みが階級的組織の中で登る必要のある一番上のレベルまで達したことを意味する。テーブルに示される場合には、その最高位レベルはシステムのレベル即ちレベル5である。割込みの非上昇フェイズの間アドレスの第1の構成部で示すレベルは、ソース及び割当て先を示す情報が循環する割込みバスのレベルとマッチする。例えば、サブシステム割込みバスがレベル3にあるとマクロクラスターを確定するアドレス構成部もレベル3である。

テーブル3

外部からチップへの割込み通信

使用中の割込みバス      ソース及び割当て先を確定する  
情報を転送するシーケンス

上昇フェイズ

クラスターバス	サブシステム、マクロクラスター、クラスター、チップ、プロセッサ
マクロクラスターバス	サブシステム、マクロクラスター、クラスター、チップ、プロセッサ
サブシステムバス	サブシステム、マクロクラスター、クラスター、チップ、プロセッサ
システムバス	サブシステム、マクロクラスター、クラスター、チップ、プロセッサ

非上昇フェイズ

サブシステムバス	マクロクラスター、クラスター、チップ、PR、サブシステム※
マクロクラスターバス	クラスター、チップ、PR、サブシステム※、マクロクラスター※
クラスターバス	チップ、PR、サブシステム※、マクロクラスター※、クラスター※

※ このアドレスは割込みの割当てを決めるのには必要ない。故に割当てバス上を送信されない。

割込みの移動が非上昇フェイズの期間中、割当て先を認定する情報のシーケンスは、割当てアドレスにある簡単な「アドレス構成部を左にシフトせよ」というオペレーションの実行によつて容易に導き出される。このことは、割込み管理回路が手渡された割込みの割当てを認定する情報のシーケンスを構成する為の簡単なアルゴリズムを構成



する為に利用される。割込み管理回路がバスを制御するようになると、第1のサイクルで割当て先の割込み管理回路が決定されることを思いだしてほしい。テーブル3で示す通り、割込みの非上昇フェイズ期間は、割当て先割込み管理回路は第1のアドレス構成部によつて容易に決定することができる。割込みの上昇フェイズ期間では、割当て先は、割当てバス上に全てゼロを送信することによつて決定する。

割込み管理回路によつて実行されるオーバーオール機能は以下の通りである。我々は、レベル1の割込み管理回路を最初念頭において説明を行う。

一方でレベル1の割込み管理回路がレベル1の割込みバスとインターフェースし、また一方ではレベル(1-1)の割込みバスともインターフェースしている事を思いだしてほしい。これらのバスは様々な方針によつて仲裁が行われる。これらの方針の典型的な例はラウンドロビン、優先性の順によるもの、位置順によるもの等である。チップ割込みバス88はラウンドロビンの順序で仲裁

が行われる。いかなる割込み管理回路も内外両側からの割込みを同時にうまく処理することが可能でなければならない。割込み管理回路と割込みバスの間のインターフェースを示すブロック図が第45図に示される。

各々の割込み管理回路は、送信されるのを待つ割込みの待ち行列を保持している。各々の割込みに関して割当てアドレスの優先順位、ランタイム情報、ソースアドレス及びフェイズ(上昇か又は非上昇か)が待ち行列になつて保持されている。割込みの上昇フェイズ期間中、関連する最高レベルから開始する全てのアドレス構成部は、ソース及び割当て先を決定する為に記憶されていなくてはならない。割込みの非上昇フェイズの期間では、割込み管理回路のレベルより低い段階であるこれらの割当てアドレス構成部だけが割当て先の決定の為に記憶される必要がある。例えばクラスター割込み管理回路218は割当ての為にチップとプロセッサを特定する情報だけを記憶する必要がある。先に入つたものを先に出す方式による待行列

の維持の方が簡単であるが、好ましい割込み管理回路は、優先順位に基づく方針で待ち行列を維持している。優先順位に基づき待ち行列の管理を行う場合、行列の中での到着時間の順にかかわらず、行列の最前列にある割込みがその列の中で最も高い優先性を持つ。

情報状況信号は、いつアドレス情報が送信されるかを示している。上記で示した通り、割当てアドレスが最初に送られ次に割込みデータ及び割込みのソースアドレスが送られる。割込み管理回路は受取つた割当てアドレスの数を計数している。この数及び割込み階級的組織に於るそのアドレスのレベルに関しわかることに基づいて、割込み管理回路はソース/割当てを決定する情報を先着順のオーダーで送るか(上昇フェイズ)又は「アドレス構成部を左に循環せよ」というオペレーションを実行するか(非上昇フェイズ)のいずれにするかを決定している。

階級的組織情報網は、RIC割込みポート機能に対する応用の一例として示されている。この割

込みポートは、割込みトポロジー及び割込みプロトコールによつてまづたく一般的なものである。階級的組織割込み情報網は、情報網のある形態にすぎない。この情報網の中では、トポロジーに関しては変化させることが可能でこのことがRICの外部割込み管理回路に影響し、2種類の変化を生む。第1の変化はレベルの数が変わると、情報網の中のアドレスの長さが変化することである。アドレスの長さが変化すると割込み管理回路内のバッファスペースも変化する。バッファスペースの変化は、RIC内に含まれていない割込み管理回路のハードウェアの設計を適当に変えることによつて操作することができる。トポロジーの変化によるもう1つの変化は、有効アドレスの組に関して起こる。この変化も、RICの外部割込み管理回路を使つてプログラムすることによつて又は他の割込み管理のハードウェアの仕様を変えることによつて操作される。

上記説明はRICのみを有する割込み情報網に限定されて示してある。しかしながら、チャネル

ル、ディスク、プリンター及び通信網インターフェースといったようなあらゆる I/O 装置を含む情報網にも同じ様に応用することができる。この応用性を保持する為に唯一必要とされることは、I/O 装置が R I C 割込みポートと接続可能なインターフェースを有しているということだけである。

以上のように当初の目的を達成し、一般的な構造を持つ I C を再構成可能にして使用者のニーズにあわせたカスタムな I C を安価に提供することができる。このような再構成可能な機能を持つことで本発明は I C により新しい大きな応用の可能性を持たせることができると確信する。

#### 4. 図面の簡単な説明

第 1 図は、本発明の再構成可能 I C の概略的全体図である。

第 2 図は、1つのプロセッサを全体的に含み各々の種類の外部インターフェースの内の 1つを含む R I C の部分的な平面図である。

第 3 図は NOR ゲート仕様として実施された D L A の AND 及び OR マトリクス部分を示す部

分図である。

第 4 図は、ゲート仕様と名前のついた実施例である D L A の AND 及び OR マトリクスの部分を示す図である。

第 5 図は制御バスの構成を示す図である。

第 6 図は、中央制御記憶制御回路と 4つのモジュール制御回路の間の関係を示す図である。

第 7 図は、中央制御記憶回路がアクセスされた場合の中央制御記憶回路及びそれぞれのモジュール制御回路のオペレーションを示すフローチャートである。

第 8 図は、花輪状に連結されたバス使用可能線の接続を示す図である。

第 9 図は各々のプロセッサ内の A L U の主要部分を示す概略図である。

第 10 図及び第 11 図は、それぞれ各々のプロセッサ内の機能ブロック及び桁上げ連結ブロック部分の回路図である。

第 12 図は各々のプロセッサ内のパレルシフタ一の構造を示すフローチャートである。

第 13 図、第 14 (a)、14 (b)、及び 14 (c) 図は、各々のプロセッサ内のパレルシフタで実行されるシフト及び／又は循環オペレーションの例を示す図であり、第 15 図はパレルシフタで実行される抽出オペレーションの例を示す図である。

第 16 図はデータバスの構造を示す図である。

第 17 図は、データバスのプロトコールを示す図である。

第 18 図は、各々のプロセッサ内のデータバスのブロック図である。

第 19 図は R I C の内部 R A M システムを概略的に示す図である。

第 20 及び 21 図はそれぞれのバス制御ユニットの構造及びオペレーションを示す図である。

第 22 図は、メモリスケデューリングユニットのオペレーションを示すフローチャートである。

第 23 図～第 25 図は隣接するプロセッサの再構成された異なるモードに対応して状況マルチプレクサが操作を行う状況バスの内部接続を示す図である。

第 26 図、第 27 図及び第 28 図は、それぞれ

独立モードロックステップモード、パイプラインモードにおける R I C 内のデータ及びコマンド命令の流れを概略的に示す図である。

第 29 図は、各々のプロセッサ内のスケデューラ及び割込み管理回路のブロック図である。

第 30 図は、各々のプロセッサ内のスケデューラのオペレーションのフローチャートである。

第 31 図は、割込み管理ワードのフォーマットを示す図である。

第 32 図は、割込みが送られる時の制御記憶アドレス線のフォーマットを示す。

第 33 図は、プロセッサ P R 3 がプロセッサ P R 1 に対し割込みを送る時の割込みのタイミングの例を示す図である。

第 34 図は、R I C のピンの配置図である。

第 35 図は、概略的に内部割込みデータ構造を示す図である。

第 36 図は、割込みが送られている間の外部割込み管理回路のオペレーションを示す図である。

第 37 図～第 41 図は、2つ以上の R I C テッ

ア上のプロセッサを連結することで可能となる構成の例を示す図である。

第42図は、4つのチップからクラスタまでの組織を示す図である。

第43図は、多数のR I Cチップが結合された階級的組織を持つシステムを示す図である。

第44図は、第43図で示した多重チップ組織に相当する樹系図である。

第45図は、第43図及び第44図で示したように階級的組織内の不特定のレベルにおける割込みレベル管理回路及びバスのオペレーションを概略的に示す図である。

第46図は、第43図及び第44図と同様の多重チップ階級的組織において割込みが上昇フェイズである時の情報プロトコールを示すタイミング図である。

第47図は、各々のプロセッサ内に含まれるマイクロシーケンサのブロック図である。

符号の説明

P R 0 ~ P R 3 … プロセッサ、1 2 … 割込み管

図面の浄書(内容に変更なし)

Fig.1

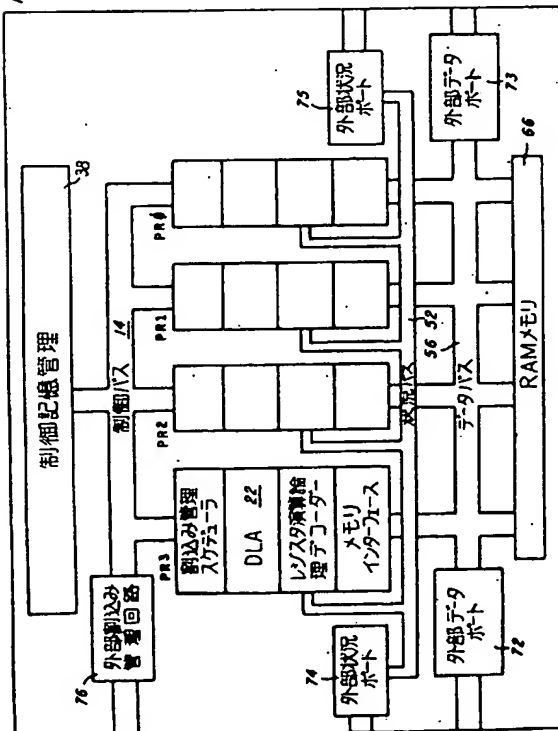


Fig.2

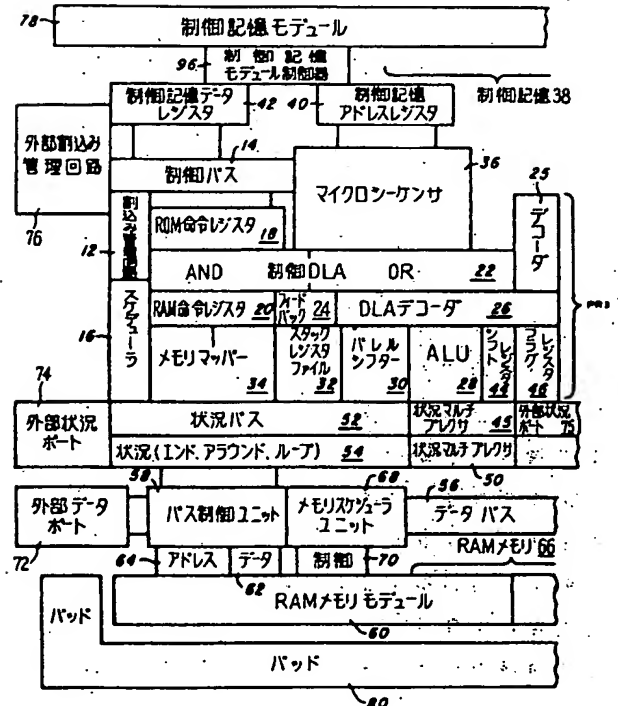


Fig. 3

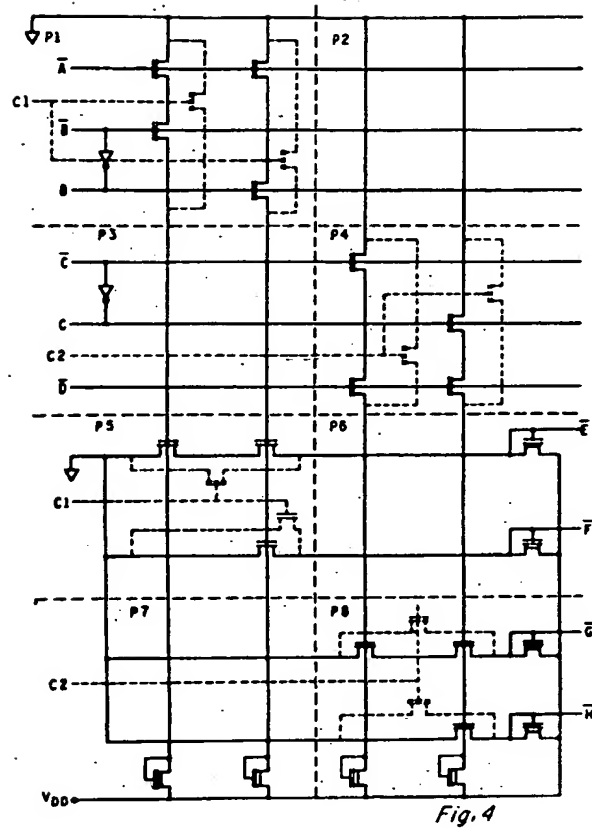
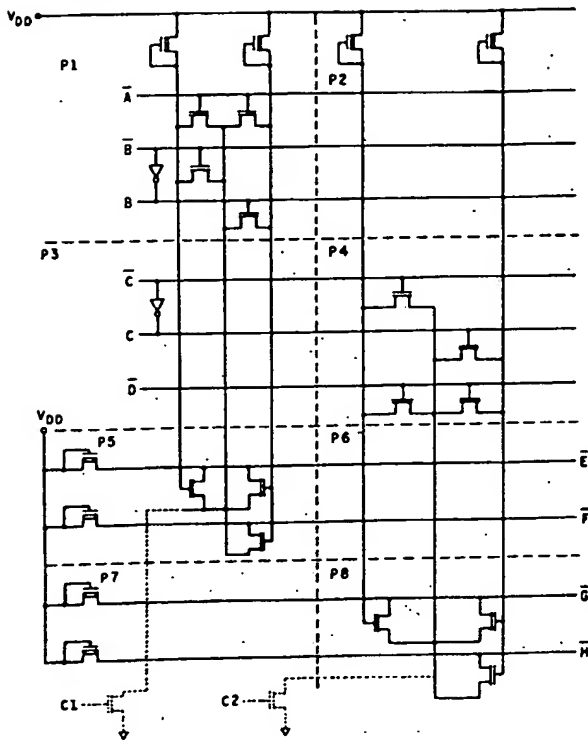


Fig. 5

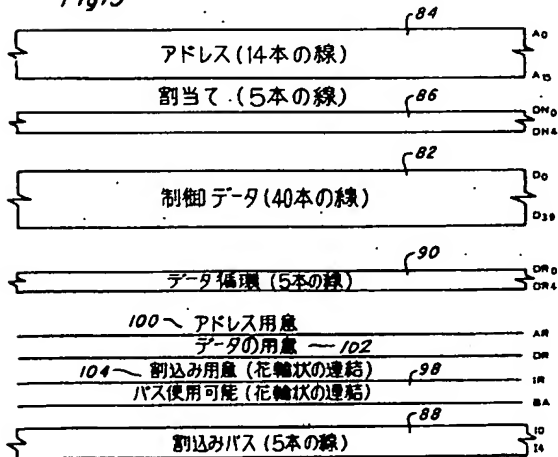
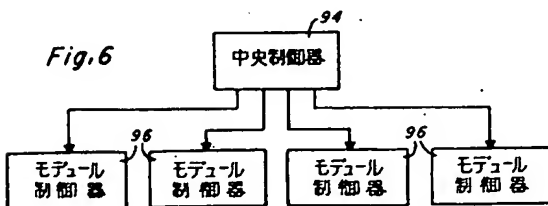


Fig. 6



中央制御器 94

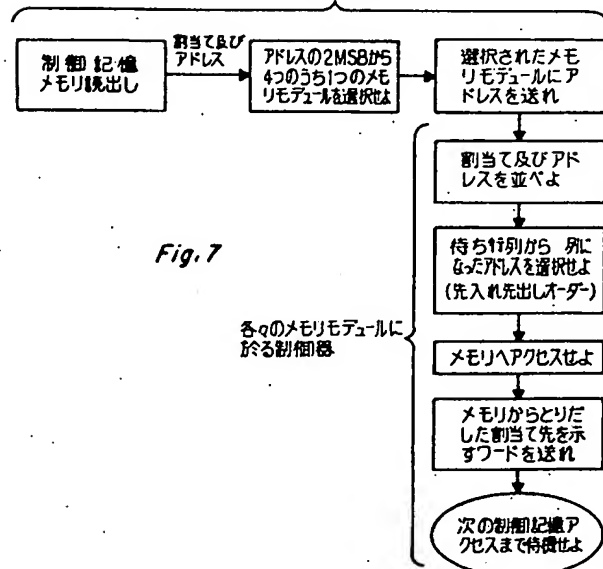


Fig. 7

各qのメモリモジュールに於ける制御器

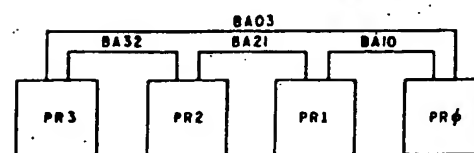
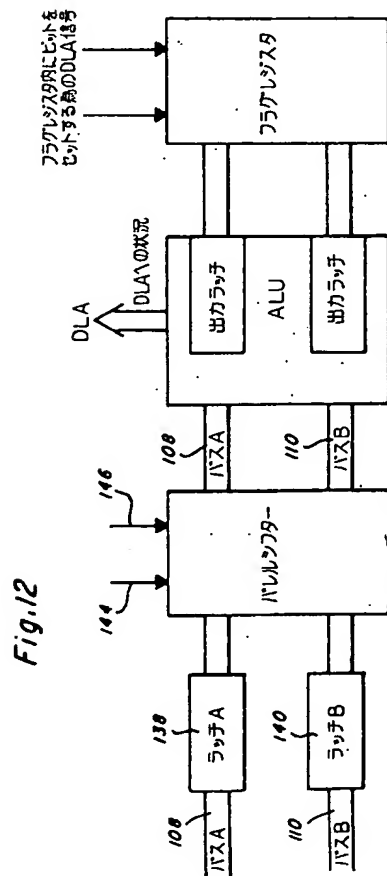
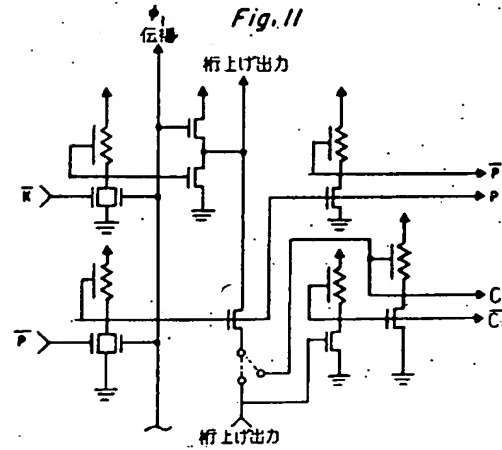
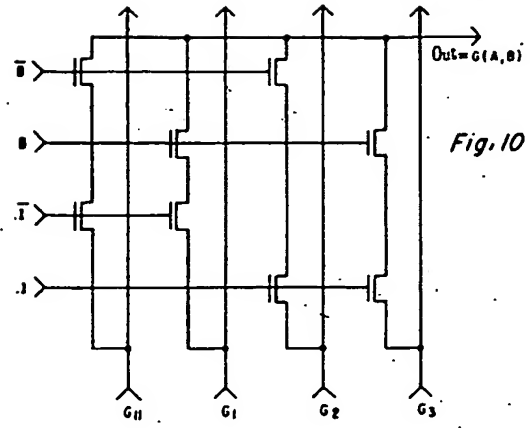
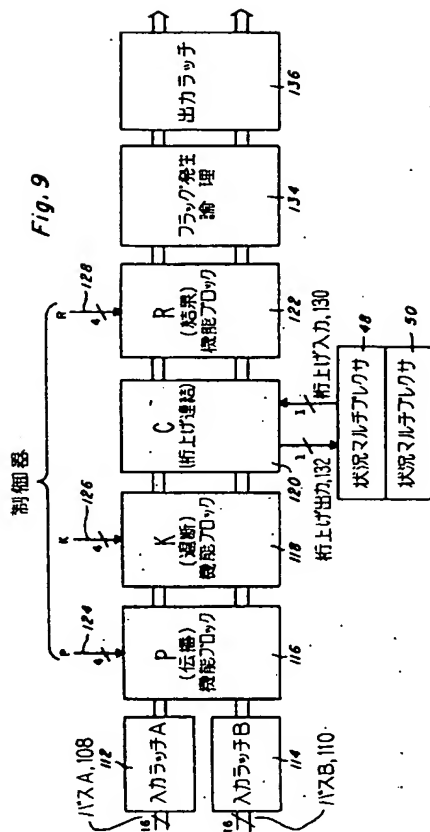
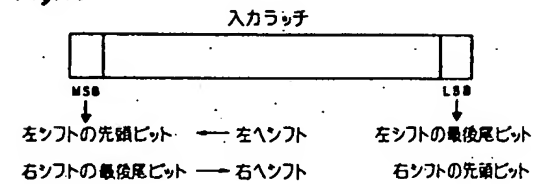


Fig. 8



**Fig. 13**



**Fig. 14**

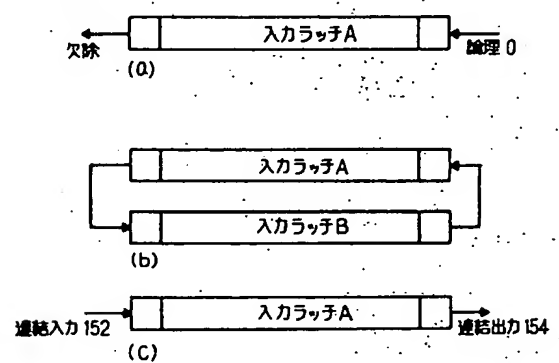


Fig.16 データバス

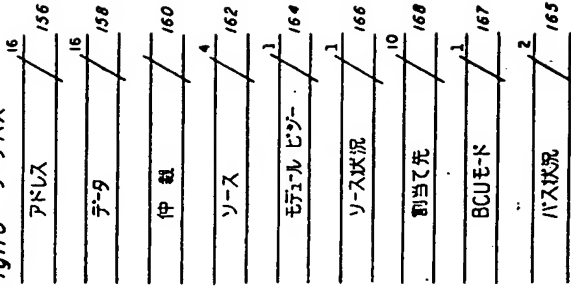


Fig.15

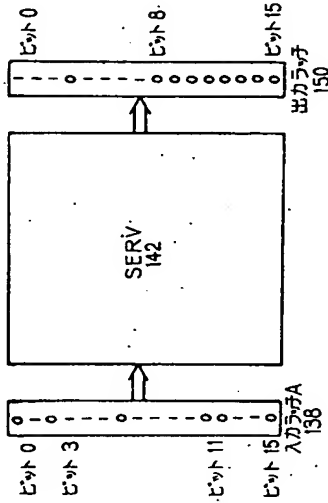


Fig.18

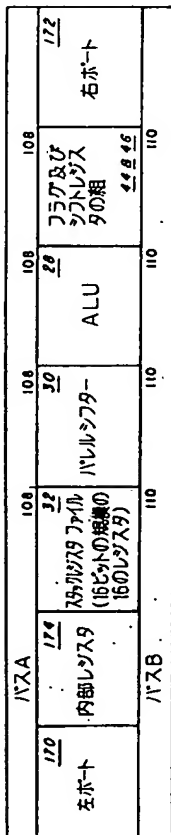


Fig.19

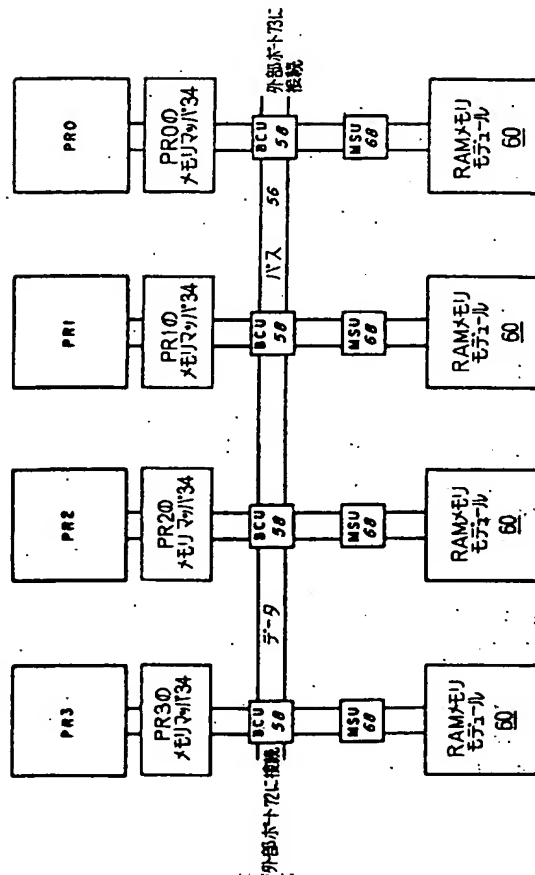
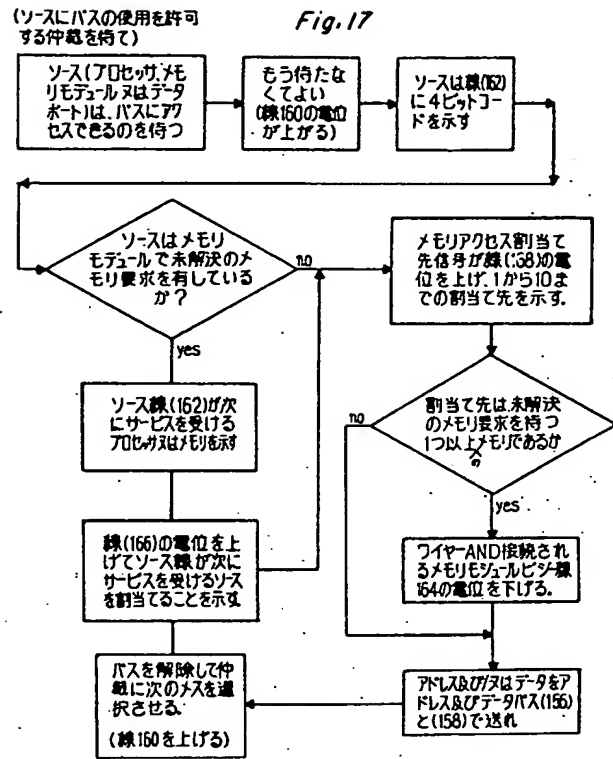
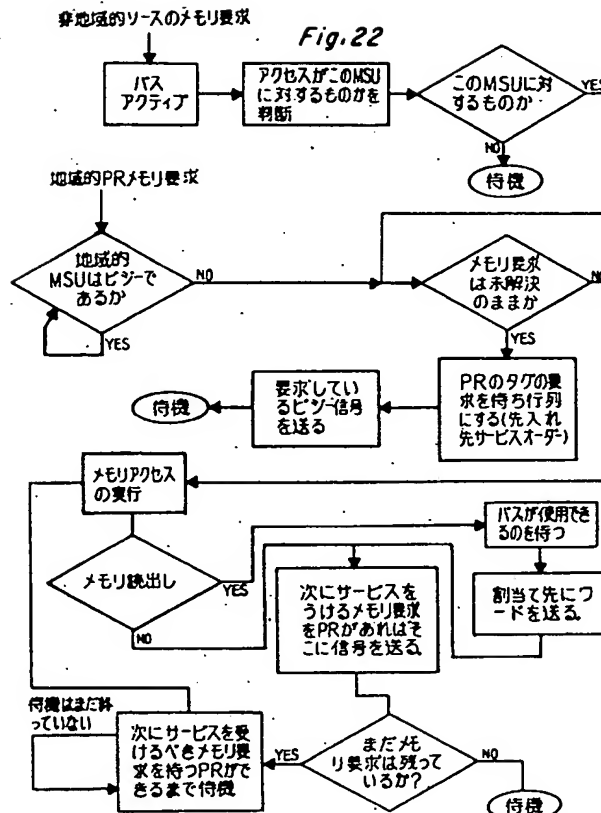
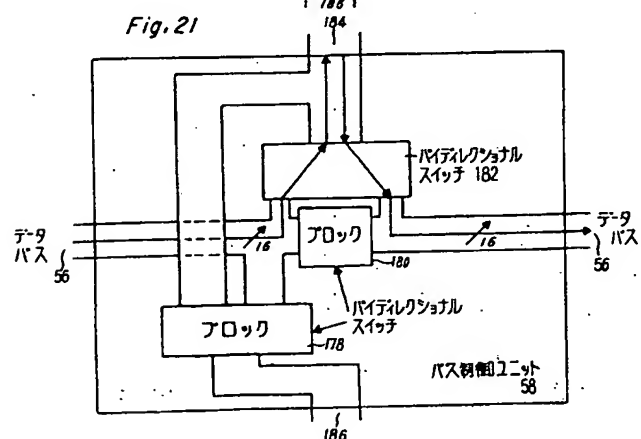
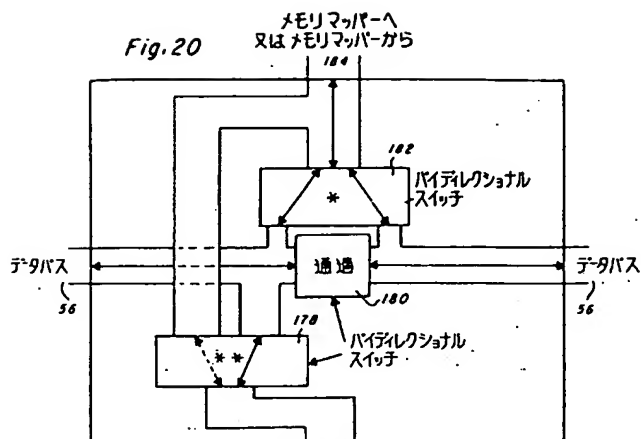


Fig.17





桁上げ状況 190	PR3	PR1
桁上げ入出力 191		
オーバーフロー 192		
負数 194		
ゼロ 196		
PR同期 198		
ソフト/循環 199		

**Fig. 23**

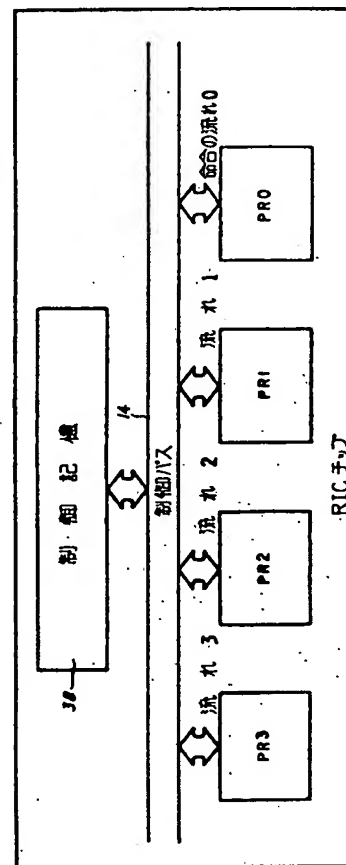
桁上げ状況 190	PR3	PR2
桁上げ入出力 191		
オーバーフロー 192		
負数 194		
ゼロ 196		
PR同期 198		
ソフト/循環 199		

**Fig. 24**

桁上げ状況 190	PR3	PR2
桁上げ入出力 191		
オーバーフロー 192		
負数 194		
ゼロ 196		
PR同期 198		
ソフト/循環 199		

**Fig. 25**

**Fig. 26**









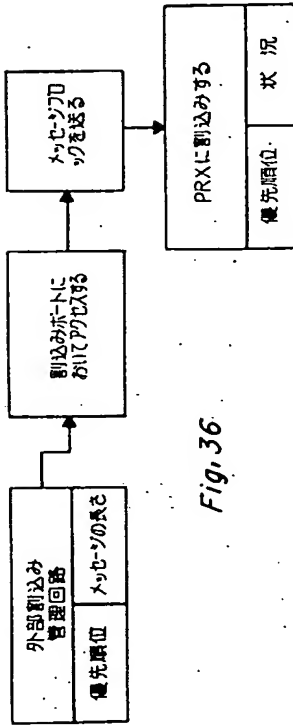


Fig. 36

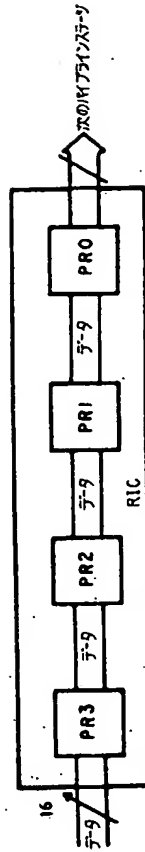


Fig. 37

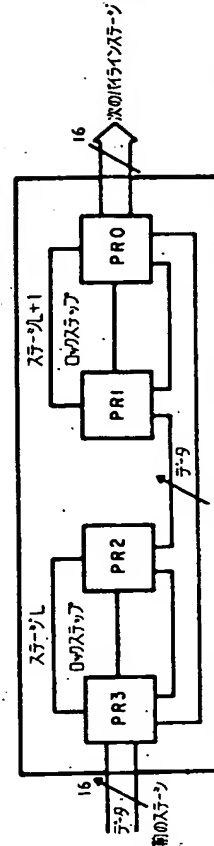


Fig. 38

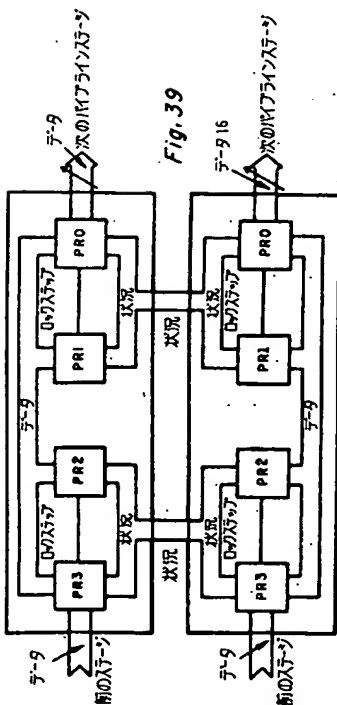


Fig. 39

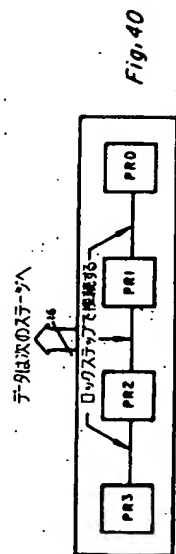


Fig. 40

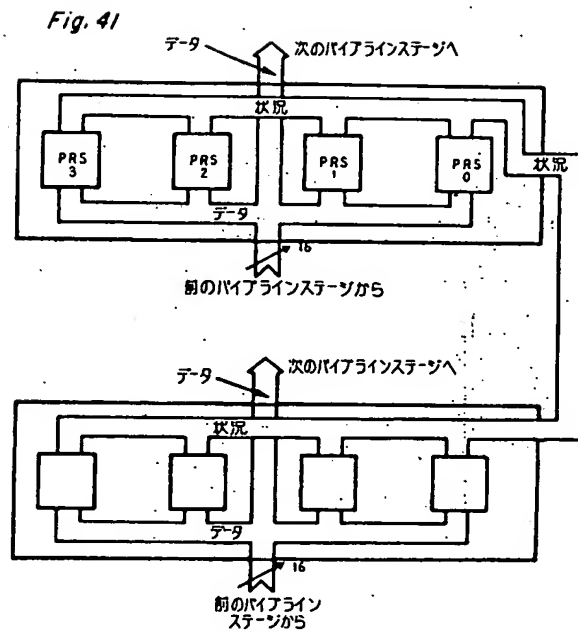
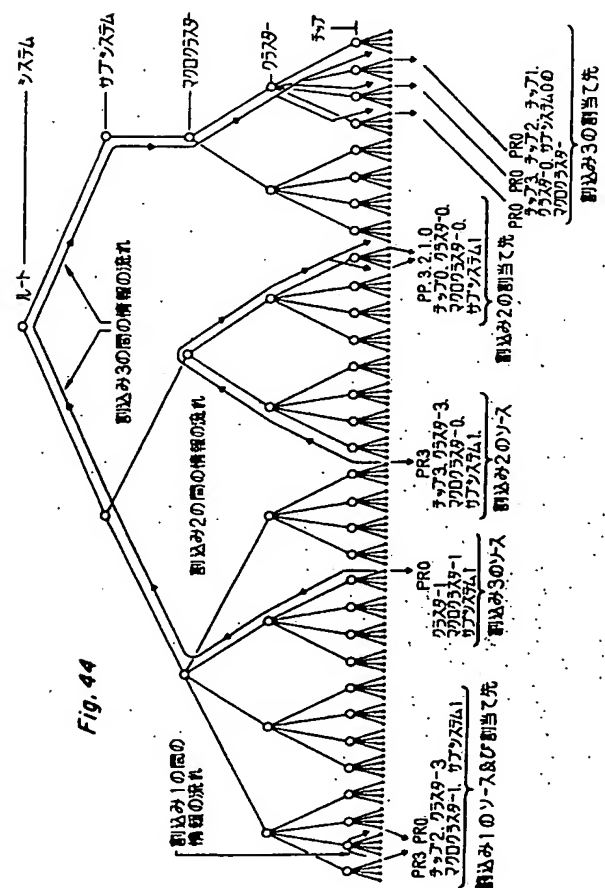
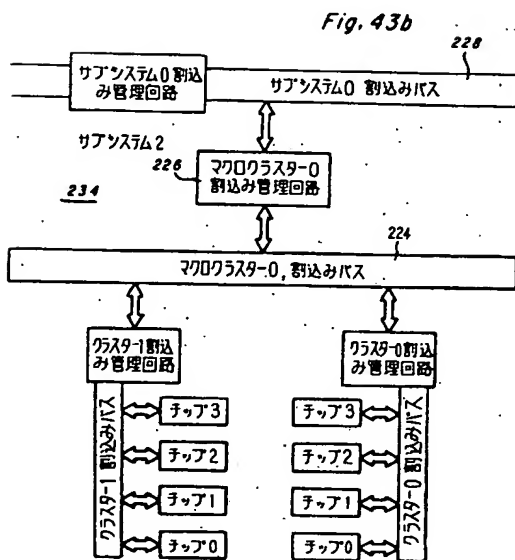
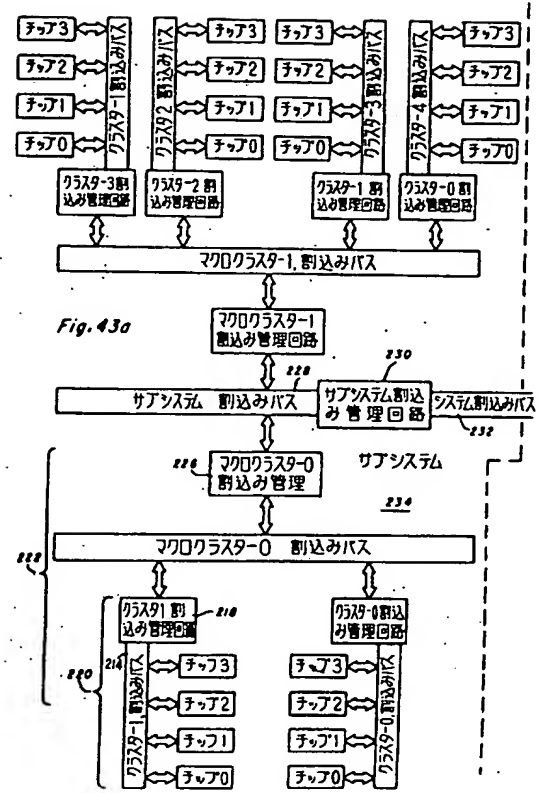
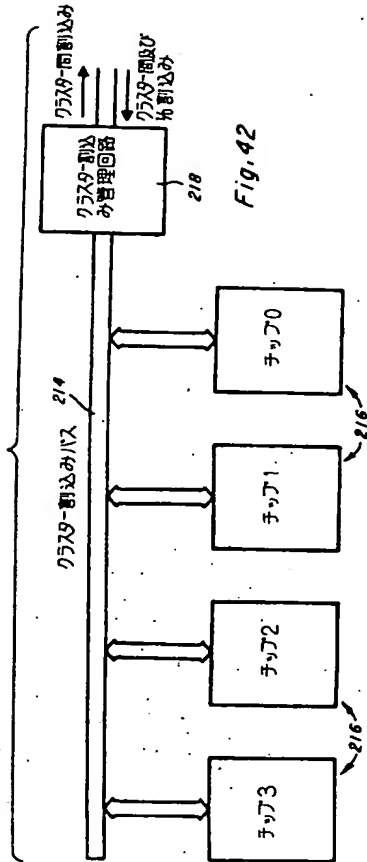
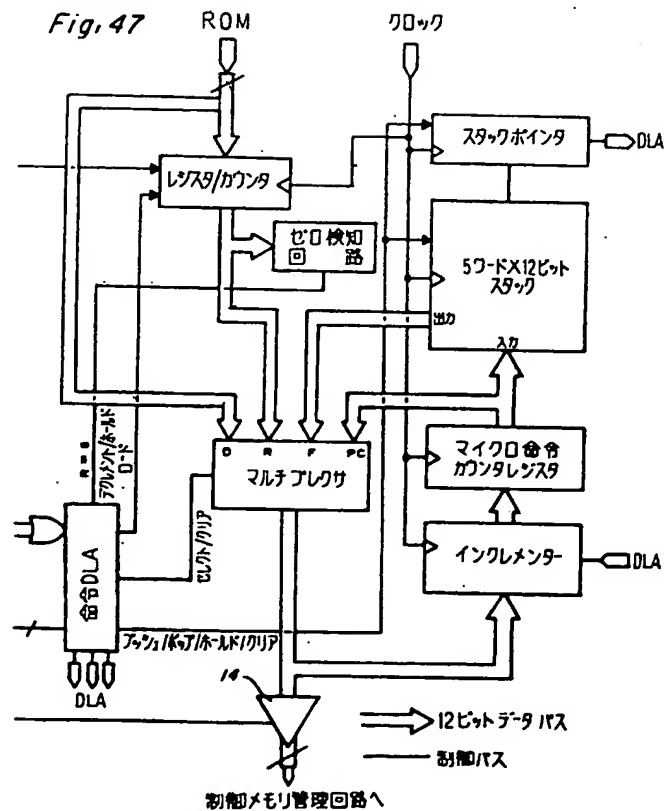
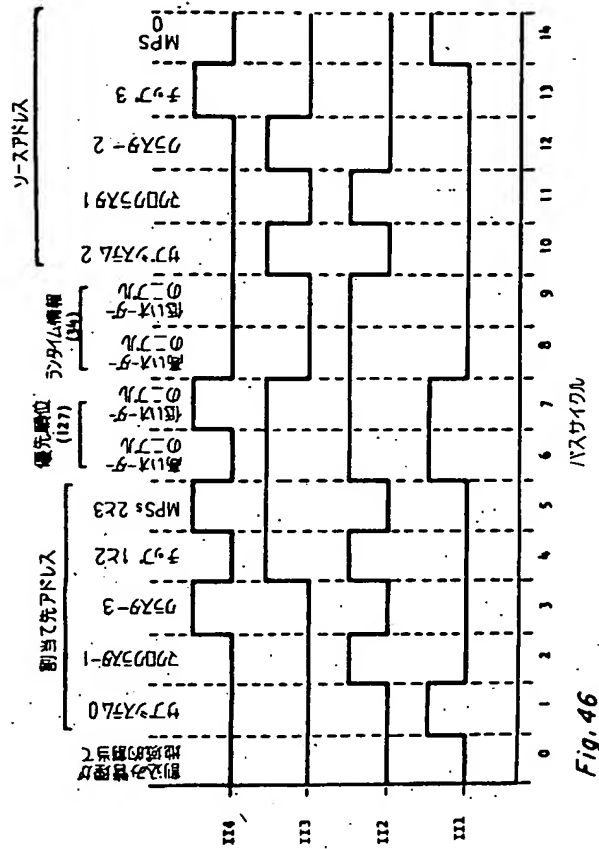
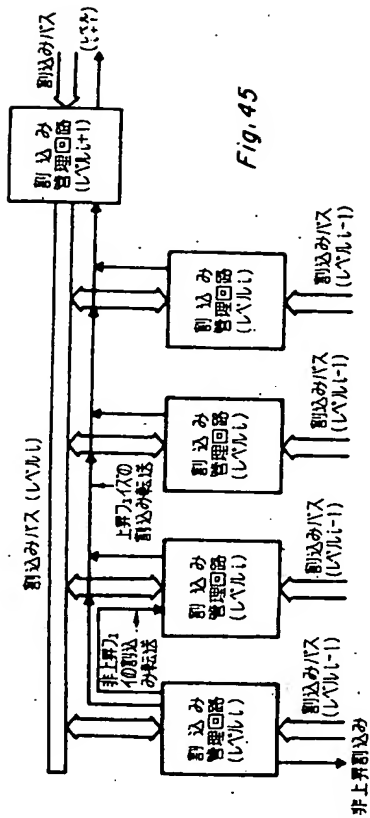


Fig. 41





手続補正書(方式) 57.10-7  
昭和 年 月 日

特許庁長官 殿

1. 事件の表示 昭和 57 年 特許願 第 128805 号

2. 発明の名称 再構成可能集積回路

3. 補正をする者

事件との関係 出願人

名称 テキサス インストルメンツ  
インコーポレーテッド

4. 代理人

住所 東京都千代田区丸の内3丁目3番1号(電話 代表 211-8741番)

氏名 (5995) 弁理士 中 村

5. 補正命令の日付 自 発

6. 補正の対象 全図面

7. 補正の内容 別紙の通り

図面の浄書(内容に変更なし)。

手続補正書 57.10-7  
昭和 年 月 日

特許庁長官 若 杉 和 夫 殿

1. 事件の表示 昭和 57 年 特許願 第 128805 号

2. 発明の名称 再構成可能集積回路

3. 補正をする者

事件との関係 出願人

名称 テキサス インストルメンツ  
インコーポレーテッド

4. 代理人

住所 東京都千代田区丸の内3丁目3番1号(電話 代表 211-8741番)

氏名 (5995) 弁理士 中 村

5. 補正命令の日付 自 発

6.

7. 補正の対象 明細書の発明の詳細な説明の欄  
および図面の簡単な説明の欄

8. 補正の内容

(1) 明細書第 / 3 頁第 / 行に「示す。」とあるを「示す。ここでビット 13 における CS が / のときは全部のコンタクトスイッチを、CS が 0 のときは部分的なコンタクトスイッチを、またビット 12 における BU が / のときはすぐにプロセスされない場合の割込みをバツファすることを、BU が 0 のときは割込みをバツファしないことをそれぞれ表わしている。」と訂正する。

(2) 同書第 / 85 頁第 / 9 行に「ある。」とあるを「あり、第 23 図は独立モード内部接続を、第 24 図はパイプラインモード内部接続を、第 25 図はロックステップモード内部接続をそれぞれ示している。」と訂正する。

(3) 同書第 / 86 頁第 7 行に「チャートである。」とあるのを次の通り訂正する。

「チャートであり、この図に用いられる、

P ( I N T . ) は割込みの優先順位を、

P ( N E X T ) は次に高い優先性をもつプロセスの優先順位を、P ( C U R ) は現在実行

中のプロセスの優先順位を、N E X T は次にスケジュールしたプロセスの優先順位を持つレジスタを、C U R R E N T は現在実行中のプロセスの優先順位を持つレジスタを、T E M P は一時的なレジスタを示している。」

(4) 明細書中、下記各個所の誤記を夫々訂正する。

頁	行	誤記	訂正
109	18	タイプ0	タイプ0(0000)
"	20	タイプ1	タイプ1(0001)
110	1	2	2(0010)
184	1	である。	であり、通常の区画を持つダイナミック論理配列を示している。
"	10	フロー	制御記憶メモリ管理のフロー
"	17	機能	ALUの機能
185	10	RIC	再構成可能IC(RIC)
186	1	独立モードロックステップモード	独立モード、ロックステップモード